ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы обработки информации и управления**»**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Б1.Б.18 Вычислительные машины системы и сети

(*код и наименование дисциплины в соответствии с РУП*)

Направление подготовки (специальность) 27.03.04 Управление в технических системах

Профиль образовательной программы Интеллектуальные системы обработки информации и управления

Форма обучения очная

Оренбург 201_ г.

СОДЕРЖАНИЕ

1. Конспект лекций.....4

1.1 Лекция №1 Общие сведения о компьютерных сетях……...4

1.2 Лекция №2 Коммутация.……8

1.3 Лекция №3 Основные характеристики ЭВМ……..15
1.4 Лекция №4 Минимальная конфигурация ЭВМ……..18
1.5 Лекция №5 Состав команд и архитектура 8086 микропроцессора……...24
1.6 Лекция №6 Сетевое оборудование……...28
1.7 Лекция №7 Протокол TCP/IP……33
1.8 Лекция №8 -9 Типовая структура процессора…….37
1.9 Лекция №10 Сетевая модель OSI……...43
1.10 Лекция №11 Запоминающие устройства ЭВМ……49
1.11 Лекция №12 Протоколы и алгоритмы маршрутизации……..….53
1.12 Лекция №13 Устройства ввода вывода информации в ЭВМ……..55
1.13 Лекция №14 Виды архитектур ЛВС……..…60
1.14 Лекция №15 Архитектура Ethernet…….65
1.15 Лекция №16 Многомашинные и многопроцессорные вычислительные системы……69
1.16 Лекция №17 Технология Token Ring……73
1.17 Лекция №18 Архитектура FDDI……74
2. Методические указания по выполнению лабораторных работ…….…….89
2.1 Лабораторная работа № ЛР-1 Сетевая модель OSI……..……89
2.2 Лабораторная работа № ЛР-2 Запоминающие устройства ЭВМ……..…102
2.3 Лабораторная работа № ЛР-3 Протоколы и алгоритмы маршрутизации……104
2.4 Лабораторная работа № ЛР-4 Устройства ввода вывода информации в ЭВМ……...108
2.5 Лабораторная работа № ЛР-5 Виды архитектур ЛВС……110
2.6 Лабораторная работа № ЛР-6 Архитектура Ethernet……..116
2.7 Лабораторная работа № ЛР-7 Многомашинные и многопроцессорные вычислительные системы……121
2.8 Лабораторная работа № ЛР-8 Технология Token Ring……..…125
2.9 Лабораторная работа № ЛР-9 Архитектура FDDI…….….129
3. Методические указания по проведению практических занятий……...…134
3.1 Практическое занятие № ПЗ-1 Общие сведения о компьютерных сетях……134
3.2 Практическое занятие № ПЗ-2 Коммутация……..….138
3.3 Практическое занятие № ПЗ-3 Основные характеристики ЭВМ……146
3.4 Практическое занятие № ПЗ-4 Минимальная конфигурация ЭВМ……...150
3.5 Практическое занятие № ПЗ-5 Состав команд и архитектура 8086 микропроцессора……...151
3.6 Практическое занятие № ПЗ-6 Сетевое оборудование……..…156
3.7 Практическое занятие № ПЗ-7 Протокол TCP/IP……...161
3.8 Практическое занятие № ПЗ-8 Типовая структура процессора……..….166

3.9 Практическое занятие № ПЗ-9 -10 Сетевая модель OSI……...168
3.10 Практическое занятие № ПЗ-11-12 Запоминающие устройства ЭВМ……175
3.11 Практическое занятие № ПЗ-13-14 Протоколы и алгоритмы маршрутизации…….180
3.12 Практическое занятие № ПЗ-15-16 Устройства ввода вывода информации в ЭВМ……185
3.13 Практическое занятие № ПЗ-17-18 Виды архитектур ЛВС……..…194
3.14 Практическое занятие № ПЗ-19-20 Архитектура Ethernet……197
3.15 Практическое занятие № ПЗ-21-22 Многомашинные и многопроцессорные вычислительные системы……..…199
3.16 Практическое занятие № ПЗ-23 Технология Token Ring……202
3.17 Практическое занятие № ПЗ-24 Архитектура FDDI……..…..206
3.18 Практическое занятие № ПЗ-25 Архитектура ATM……..……..208

1. Конспект лекций

1.1 Лекция №1. Общие сведения о компьютерных сетях (2 часа)

1.1.1 Вопросы лекции:

1. Классификация сетей.
2. Топология сетей.

1.1.2 Краткое содержание вопросов:

1. Классификация сетей.

Классификация сетей ЭВМ (компьютерных сетей), как любых больших и сложных систем, может быть выполнена на основе различных признаков, в качестве которых могут быть использованы (рис.1):

- размер (территориальный охват) сети;
- принадлежность;
- назначение;
- область применения.

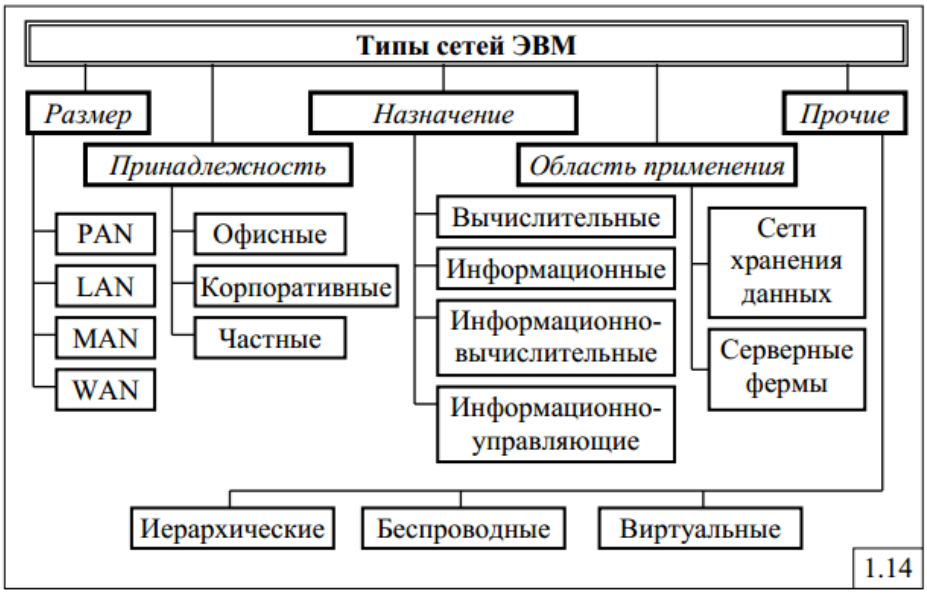


Рисунок 1.Типы сетей ЭВМ

1. **По размеру (территориальному охвату)** сети ЭВМ делятся на:

- персональные;
- локальные;
- городские (региональные).
- глобальные.

Персональная сеть (PersonalAreaNetwork, PAN) — это сеть,объединяющая персональные электронные устройства пользователя (телефоны, карманные персональные компьютеры, смартфоны, ноутбуки и т.п.) и характеризующаяся:

- небольшим числом абонентов;
- малым радиусом действия (до нескольких десятков метров);
- некритичностью к отказам.

К стандартам таких сетей в настоящее время относятся Bluetooth, Zigbee, Пиконет.

Локальная вычислительная сеть (ЛВС) (LocalAreaNetwork, LAN)– сеть со скоростью передачи данных , как правило, не менее 1 Мбит/с, обеспечивающая связь на небольших расстояниях – от нескольких десятков метров до нескольких километров. Оборудование, подключаемое к ЛВС, может находиться в одном или нескольких соседних зданиях.

ПримерыЛВС: Ethernet, Token Ring.

Городская вычислительная сеть (MetropolitanAreaNetwork, MAN) – сеть, промежуточная по размеру между ЛВС и глобальной сетью.

Протоколы и кабельная система для городской вычислительной сети описываются в стандартах комитета IEEE 802.6. MAN реализуется на основе протокола DQDB (DistributedQueueDualBus) – двойная шина сраспределенной очередью и использует волоконно-оптический кабель для передачи данных со скоростью 100 Мбит/с на территории до 100 км2. MAN может применяться для объединения в одну сеть группы сетей, расположенных в разных зданиях. Последние разработки, связанные с высокоскоростным

беспроводным доступом в соответствии со стандартом IEEE 802.16, привели к созданию MAN в виде широкополосных беспроводных ЛВС.

Глобальная сеть (WideAreaNetwork, WAN) – в отличие от ЛВС охватывает большую территорию и представляет собой объединение нескольких ЛВС, связанных с помощью специального сетевого оборудования (маршрутизаторов, коммутаторов и шлюзов), образующих в случае использования высокоскоростных каналов магистральную сеть передачи данных (магистральную сеть связи). Наиболее широкое применение находят глобальные сети для нужд информационного обмена в коммерческих, научных и других профессиональных целях.

Для построения глобальных сетей могут использоваться различные сетевые технологии, в том числе TCP/IP, X.25, FrameRelay, ATM, MPLS.

Настоящей глобальной сетью, пожалуй, можно считать только сеть Интернет. Вряд ли глобальной можно считать сеть, объединяющую 2-3 ЛВС, находящиеся в разных городах, расположенных на расстоянии нескольких десятков или даже сотен километров друг от друга. Однако, поскольку для построения такой «простой» сети используются обычно те же сетевые технологии и технические средства, что и в сети Интернет, то такие сети обычно тоже относят к классу глобальных сетей.

2. **По принадлежности** сети ЭВМ делятся на:

- офисные – сети, расположенные на территории офиса компании, ограниченной обычно пределами одного здания, и построенные на технологиях LAN;
- корпоративные (ведомственные) – сети, представляющие собой объединение нескольких офисных сетей компании, расположенных в разных территориально разнесенных зданиях, находящихся возможно в разных городах и регионах, и построенные на технологиях MAN или WAN;
- частные – сети, построенные обычно на технологии виртуальной частной сети (VirtualPrivateNetwork, VPN), позволяющей обеспечить одно или несколько сетевых соединений, которые могут быть трёх видов: узел-узел, узел-сеть и сеть-сеть, образующих логическую сеть поверхдругой сети (например, Интернет).

3. **По назначению** сети ЭВМ делятся на:

- вычислительные, предназначенные для решения задач пользователей, ориентированных, в основном, на вычисления;
- информационные, ориентированные на предоставление информационных услуг; примерами таких сетей могут служить сети, предоставляющие справочные и библиотечные услуги;

2. Топология сетей.

Термин ***топология сети*** означает способ соединения компьютеров в сеть. Вы также можете услышать другие названия – ***структура сети*** или***конфигурация сети*** (это одно и то же). Кроме того, понятие топологии включает множество правил, которые определяют места размещения компьютеров, способы прокладки кабеля, способы размещения связующего оборудования и многое другое. На сегодняшний день сформировались и устоялись несколько основных топологий. Из них можно отметить “**шину**”, “**кольцо**” и “**звезду**”.

Топология “шина”

Топология ***шина*** (рис.2) (или, как ее еще часто называют ***общая шина*** или ***магистраль***) предполагает использование одного кабеля, к которому подсоединены все рабочие станции.

Рисунок 2. Топология шина.

Общий кабель используется всеми станциями по очереди. Все сообщения, посылаемые отдельными рабочими станциями, принимаются и прослушиваются всеми остальными компьютерами, подключенными к сети. Из этого потока каждая рабочая станция отбирает адресованные только ей сообщения.

Достоинства топологии “шина”:

- простота настройки;
- относительная простота монтажа и дешевизна, если все рабочие станции расположены рядом;
- выход из строя одной или нескольких рабочих станций никак не отражается на работе всей сети.

Недостатки топологии “шина”:

- неполадки шины в любом месте (обрыв кабеля, выход из строя сетевого коннектора) приводят к неработоспособности сети;
- сложность поиска неисправностей;
- низкая производительность – в каждый момент времени только один компьютер может передавать данные в сеть, с увеличением числа рабочих станций производительность сети падает;
- плохая масштабируемость – для добавления новых рабочих станций необходимо заменять участки существующей шины.

Именно по топологии “шина” строились локальные сети на коаксиальном кабеле. В этом случае в качестве шины выступали отрезки коаксиального кабеля, соединенные Т-коннекторами. Шина прокладывалась через все помещения и подходила к каждому компьютеру. Боковой вывод Т-коннектора вставлялся в разъем на сетевой карте. Сейчас такие сети безнадежно устарели и повсюду заменены “звездой” на витой паре, однако оборудование под коаксиальный кабель еще можно увидеть на некоторых предприятиях.

Топология “кольцо”

Кольцо – это топология локальной сети, в которой рабочие станции подключены последовательно друг к другу, образуя замкнутое кольцо. Данные передаются от одной рабочей станции к другой в одном направлении (по кругу) (рис.3). Каждый ПК работает как повторитель, ретранслируя сообщения к следующему ПК, т.е. данные передаются от одного компьютера к другому как бы по эстафете.

Рисунок 3. Топология кольцо.

Если компьютер получает данные, предназначенные для другого компьютера – он передает их дальше по кольцу, в ином случае они дальше не передаются.

Достоинства кольцевой топологии:

- простота установки;
- практически полное отсутствие дополнительного оборудования;
- возможность устойчивой работы без существенного падения скорости передачи данных при интенсивной загрузке сети.

Однако “кольцо” имеет и существенные недостатки:

- каждая рабочая станция должна активно участвовать в пересылке информации; в случае выхода из строя хотя бы одной из них или обрыва кабеля – работа всей сети останавливается;
- подключение новой рабочей станции требует краткосрочного выключения сети, поскольку во время установки нового ПК кольцо должно быть разомкнуто;
- сложность конфигурирования и настройки;
- сложность поиска неисправностей.

Кольцевая топология сети используется довольно редко. Основное применение она нашла в оптоволоконных сетях стандарта TokenRing.

Топология “звезда”

Звезда – это топология локальной сети, где каждая рабочая станция присоединена к центральному устройству (коммутатору или маршрутизатору) (рис.4). Центральное устройство управляет движением пакетов в сети. Каждый компьютер через сетевую карту подключается к коммутатору отдельным кабелем.

Рисунок 4. Топология звезда.

При необходимости можно объединить вместе несколько сетей с топологией “звезда” – в результате вы получите конфигурацию сети с ***древовидной*** топологией. Древовидная топология распространена в крупных компаниях. Мы не будем ее подробно рассматривать в данной статье.

Топология “звезда” на сегодняшний день стала основной при построении локальных сетей. Это произошло благодаря ее многочисленным достоинствам:

- выход из строя одной рабочей станции или повреждение ее кабеля не отражается на работе всей сети в целом;
- отличная масштабируемость: для подключения новой рабочей станции достаточно проложить от коммутатора отдельный кабель;
- легкий поиск и устранение неисправностей и обрывов в сети;
- высокая производительность;
- простота настройки и администрирования;
- в сеть легко встраивается дополнительное оборудование.
-

Однако, как и любая топология, “звезда” не лишена недостатков:

- выход из строя центрального коммутатора обернется неработоспособностью всей сети;
- дополнительные затраты на сетевое оборудование – устройство, к которому будут подключены все компьютеры сети (коммутатор);
- число рабочих станций ограничено количеством портов в центральном коммутаторе.

Звезда – самая распространенная топология для проводных и беспроводных сетей. Примером звездообразной топологии является сеть с кабелем типа витая пара, и коммутатором в качестве центрального устройства. Именно такие сети встречаются в большинстве организаций.

1.2 Лекция №2. Коммутация (2 часа)

1.2.1 Вопросы лекции:

1. Способы коммутации.
2. Разделение каналов по времени.
3. Разделение каналов по частоте.

1.2.2 Краткое содержание вопросов:

1. Способы коммутации.

Пакеты в сети могут передаваться двумя способами (рис.5):

- дейтаграммным;
- путем формирования «виртуального канала».

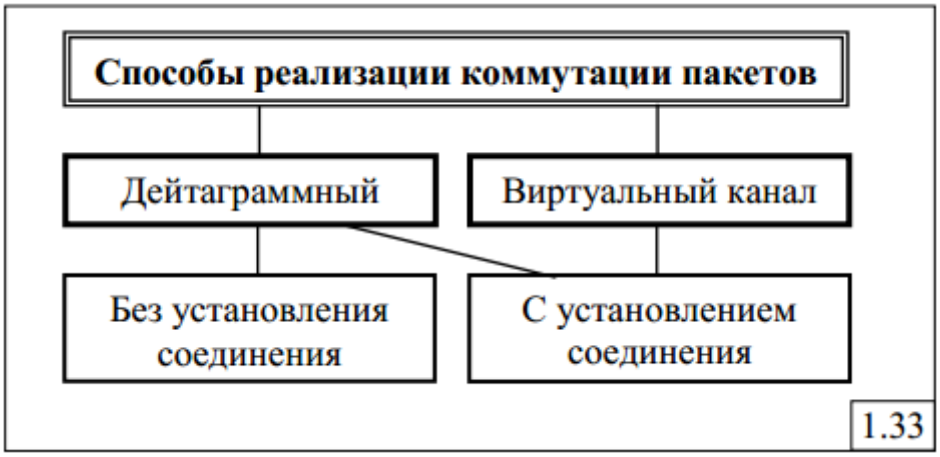


Рисунок 5. Способы реализации коммутации.

При **дейтаграммном** способе пакеты одного и того же сообщения могут передаваться между двумя взаимодействующими пользователями А и В по разным маршрутам, как это показано на рис.6, где пакет П1 передаётся по маршруту У1-У2-У6-У7, пакет П2 – по маршруту У1-У4-У7 и пакет П3 – по маршруту У1-У3-У5-У7. В результате такого способа передачи все пакеты приходят в конечный узел сети в разное время и в произвольной последовательности. Пакеты одного и того же сообщения, рассматриваемые в каждом узле сети как самостоятельные независимые единицы данных и передаваемые разными маршрутами, называются дейтаграммами (datagram). В узлах сети для каждой дейтаграммы всякий раз определяется наилучший путь передачи в соответствии с выбранной метрикой маршрутизации, не зависимо от того, по какому пути переданы были предыдущие дейтаграммы с такими же адресами назначения (получателя) и источника (отправителя).

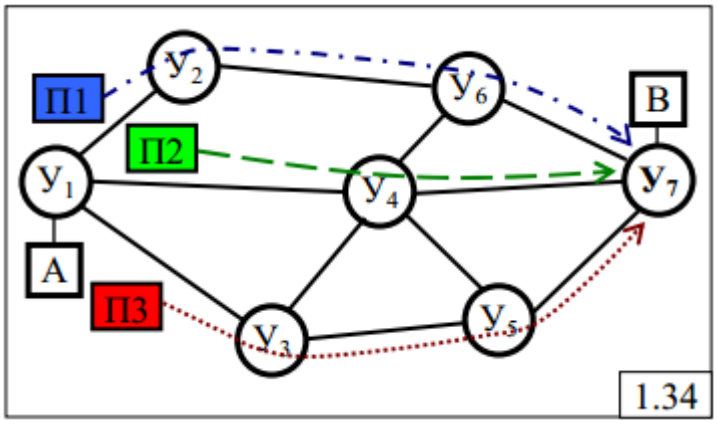


Рисунок 6. Дейтаграммный способ.

Дейтаграммный способ передачи пакетов может быть реализован:

- без установления соединения между абонентами сети;
- с установлением соединения между взаимодействующими абонентами сети.

В последнем случае между взаимодействующими абонентами предварительно устанавливается соединение путём обмена служебными пакетами: «запрос на соединение» и «подтверждение соединения», означающее готовность принять передаваемые данные. В процессе установления соединения могут «оговариваться» значения параметров передачи данных, которые должны выполняться в течение сеанса связи. После установления соединения отправитель начинает передачу, причём пакеты одного и того же сообщения могут передаваться разными маршрутами, то есть дейтаграммным способом. По завершении сеанса передачи данных выполняется процедура разрыва соединения путём обмена служебными пакетами: «запрос на разрыв соединения» и «подтверждение разрыва соединения». Описанная процедура передачи пакетов с установлением соединения иллюстрируется на диаграмме (рис.7).

Достоинствами дейтаграммного способа передачи пакетов в компьютерных сетях являются:

- простота организации и реализации передачи данных – каждый пакет (дейтаграмма) сообщения передаётся независимо от других пакетов;
- в узлах сети для каждого пакета выбирается наилучший путь (маршрут);
- передача данных может выполняться как без установления соединения между взаимодействующими абонентами, так и при необходимости с установлением соединения.

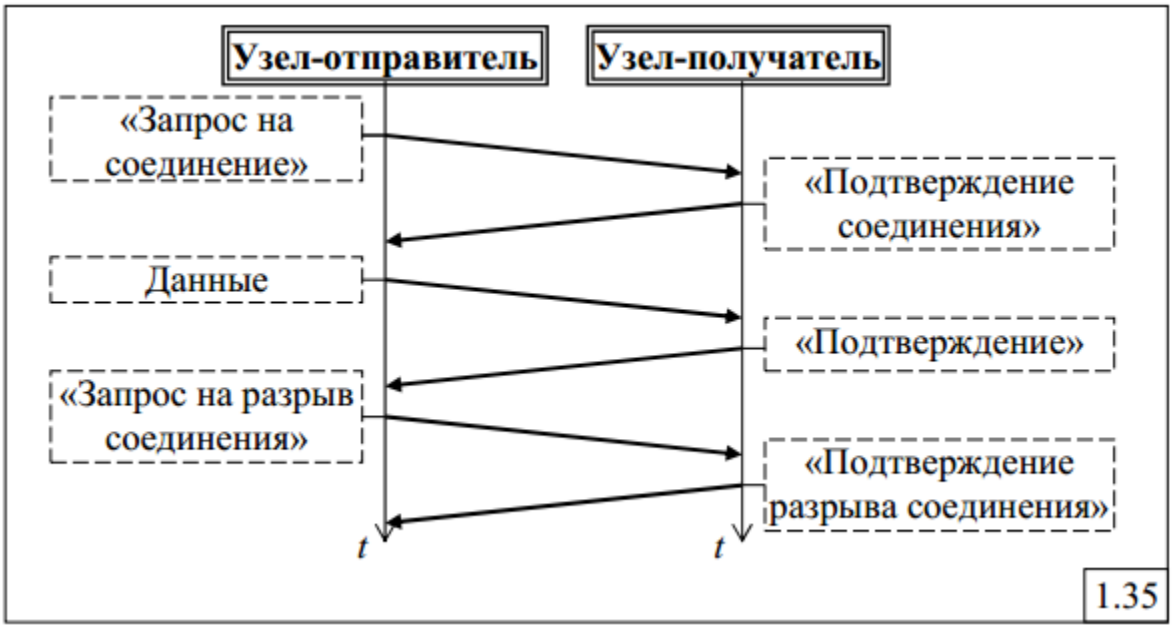


Рисунок 7. Процедура передачи пакетов

К недостаткам дейтаграммного способа передачи пакетов следует отнести:

- необходимость сборки сообщения в конечном узле: сообщение не может быть передано получателю, пока в конечном узле сети не соберутся все пакеты данного сообщения, поэтому в случае потери хотя бы одного пакета сообщение не сможет быть сформировано и передано получателю;
- при длительном ожидании пакетов одного и того же сообщения в конечном узле может скопиться достаточно большое количество пакетов сообщений, собранных не

полностью, что требует значительных затрат на организацию в узле буферной памяти большой ёмкости;

- для предотвращения переполнения буферной памяти узла время нахождения (ожидания) пакетов одного и того же сообщения в конечном узле ограничивается, и по истечении этого времени все поступившие пакеты не полностью собранного сообщения уничтожаются, после чего выполняется запрос на повторную передачу данного сообщения; это приводит к увеличению нагрузки на сеть и, как следствие, к снижению её производительности, измеряемой количеством сообщений, передаваемых в сети за единицу времени.

Способ передачи пакетов «**виртуальный канал**» заключается в формировании единого «виртуального» канала на время взаимодействия абонентов для передачи всех пакетов сообщения. Этот способ реализуется с использованием предварительного установления соединения между взаимодействующими абонентами, в процессе которого формируется наиболее рациональный единый для всех пакетов маршрут, по которому, в отличие от дейтаграммного способа, все пакеты сообщения передаются в естественной последовательности, как это показано на рис.8.

Пакеты П1, П2 и П3 сообщения передаются в естественной последовательности от пользователя А к пользователю В по предварительно созданному виртуальному каналу через узлы У1-У4-У7.

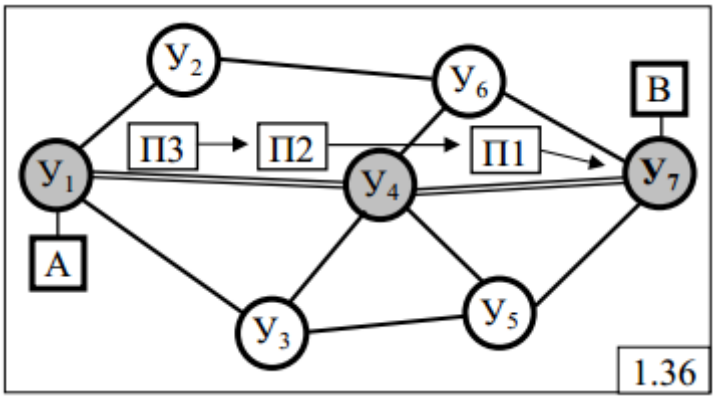


Рисунок 8. Передача пакетов сообщения.

Виртуальный канал, как и реальный физический канал в случае коммутации каналов, существует только в течение сеанса связи, при этом ресурсы реальных каналов связи (пропускная способность) и узлов сети (буферная память), находящихся на маршруте, резервируются на всё время сеанса.

Не следует путать и смешивать коммутацию каналов и способ передачи пакетов «виртуальный канал». Основное их отличие состоит в том, что «виртуальной канал» реализуется с промежуточным хранением пакетов в узлах сети, в то время как коммутация каналов реализуется без промежуточного хранения передаваемых пакетов за счёт создания реального (а не виртуального) физического канала между абонентами сети.

К достоинствам способа передачи пакетов «виртуальный канал» по сравнению с дейтаграммной передачей пакетов можно отнести:

- меньшие задержки в узлах сети, обусловленные резервированием ресурсов, и прежде всего пропускной способности каналов связи, в процессе установления соединения;
- небольшое время ожидания в конечном узле для сборки всего сообщения, поскольку пакеты передаются последовательно друг за другом по одному и тому же маршруту (виртуальному каналу), и вероятность того, что какой-либо пакет «заблудится» в результате неудачно выбранногомаршрута или его время доставки окажется слишком большим, как это может произойти при дейтаграммном способе, близка к нулю;

• более эффективное использование буферной памяти промежуточных узлов за счёт её предварительного резервирования, а также буферной памяти в конечном узле в связи с небольшим временем ожидания прихода всех пакетов сообщения.

К недостаткам способа передачи пакетов «виртуальный канал» можно отнести:

• наличие накладных расходов (издержек) на установление соединения;

• неэффективное использование ресурсов сети, поскольку они резервируются на всё время взаимодействия абонентов (сеанса) и не могут быть предоставлены другому соединению, даже если они в данный момент не используются.

2. Разделение каналов по времени.

Коммутаторы, а также соединяющие их каналы должны обеспечивать одновременную передачу данных нескольких абонентских каналов. Для этого они должны быть высокоскоростными и поддерживать какую-либо технику мультиплексирования абонентских каналов. В настоящее время для мультиплексирования абонентских каналов используются две техники: техника частотного мультиплексирования (Frequency Division Multiplexing, FDM); техника мультиплексирования с разделением времени (Time Division Multiplexing, TDM).

Коммутация каналов на основе частотного мультиплексирования

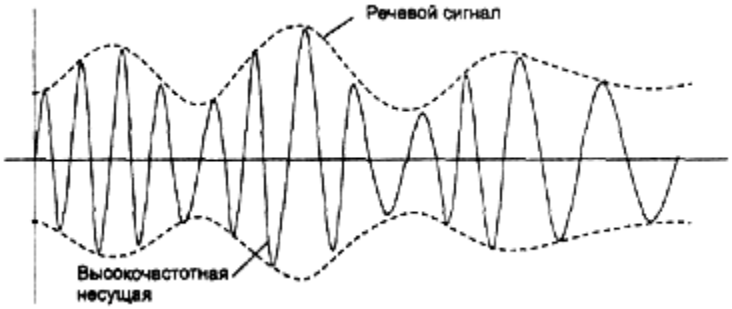


Рисунок 9. Модуляция речевых сигналов

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

Для разделения абонентских каналов характерна техника модуляции высокочастотного несущего синусоидального сигнала низкочастотным речевым сигналом (рис. 9). Если сигналы каждого абонентского канала перенести в свой собственный диапазон частот, то в одном широкополосном канале можно одновременно передавать сигналы нескольких абонентских каналов. На входы FDM- коммутатора поступают исходные сигналы от абонентов телефонной сети. Коммутатор выполняет перенос частоты каждого канала в свой диапазон частот. Обычно высокочастотный диапазон делится на полосы, которые отводятся для передачи данных абонентских каналов. Такой канал называют уплотненным. Коммутаторы FDM могут выполнять как динамическую, так и постоянную ком¬мутацию. При динамической коммутации один абонент инициирует соединение с другим абонентом, посылая в сеть номер вызываемого абонента. Коммутатор дина¬мически выделяет данному абоненту одну из свободных полос своего уплотненно¬го канала. При постоянной коммутации за абонентом полоса закрепляется на длительный срок путем настройки коммутатора по отдельному входу, недоступ¬ному пользователям. Коммутация каналов на основе разделения времени разрабатывалась в расчете на передачу непрерывных сигналов, представляющих голос. При переходе к цифро¬вой форме представления голоса была разработана новая техника мультиплекси¬рования, ориентирующаяся на дискретный характер передаваемых данных. Эта техника носит название мультиплексирования с разделением времени (Time Division Multiplexing, TDM). Реже используется и другое ее название — техника синхронного режима

передачи (Synchronous Transfer Mode, STM). Аппаратура TDM-сетей — мультиплексоры, коммутаторы, демультиплексоры — работает в режиме разделения времени, поочередно обслуживая в течение цикла своей работы все абонентские каналы. Цикл работы оборудования TDM равен 125 мкс, что соответствует периоду следования замеров голоса в цифровом абонентском канале. Это значит, что мультиплексор или коммутатор успевает вовремя обслужить любой абонентский канал и передать его очередной замер далее по сети. Каждому соединению выделяется один квант времени цикла работы аппаратуры, называемый также тайм-слотом. Длительность тайм-слота зависит от числа абонентских каналов, обслуживаемых мультиплексором TDM или коммутатором. Мультиплексор принимает информацию по N входным каналам от конечных абонентов, каждый из которых передает данные по абонентскому каналу со скоростью 64 Кбит/с — 1 байт каждые 125 мкс. В каждом цикле мультиплексор выполняет следующие действия: прием от каждого канала очередного байта данных; составление из принятых байтов уплотненного кадра, называемого также обоймой; передача уплотненного кадра на выходной канал с битовой скоростью, равной Nx64 Кбит/с. Порядок байт в обойме соответствует номеру входного канала, от которого этот байт получен. Количество обслуживаемых мультиплексором абонентских каналов зависит от его быстродействия. Демультиплексор выполняет обратную задачу — он разбирает байты уплотненного кадра и распределяет их по своим нескольким выходным каналам, при этом он считает, что порядковый номер байта в обойме соответствует номеру выходного канала.

Коммутатор принимает уплотненный кадр по скоростному каналу от мульти¬плексора и записывает каждый байт из него в отдельную ячейку своей буферной памяти, причем в том порядке, в котором эти байты были упакованы в уплотненный кадр. Для выполнения операции коммутации байты извлекаются из буферной памяти не в порядке поступления, а в таком порядке, который соответствует под¬держиваемым в сети соединениям абонентов.

Развитием идей статистического мультиплексирования стала технология асинхронного режима передачи — ATM, которая вобрала в себя лучшие черты техники коммутации каналов и пакетов.

3.Разделение каналов по частоте

Частотное разделение каналов (ЧРК) — разделение каналов по частоте, при котором каждому каналу выделяется определённый диапазон частот. В многоканальных системах связи (МКС) с ЧРК канальные сигналы отличаются друг от друга положением своих спектров на оси частот. Обычно системы с ЧРК используются для передачи аналоговых сигналов. На рис. 10 представлена структурная схема простейшей МКС с ЧРК.

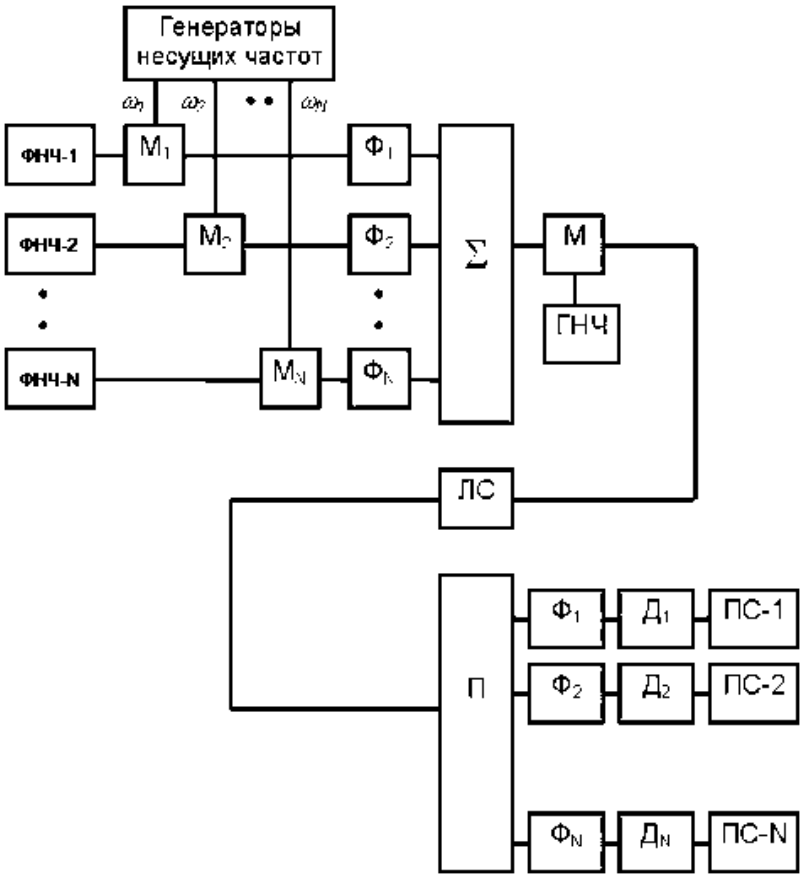


Рисунок 10. Структурная схема многоканальной системы с ЧРК

Наиболее часто в отдельных каналах при ЧРК применяется однополосная модуляция с соответственно подобранными частотами пилот-сигналов, которые выдаются генератором несущих частот (ГНЧ). Данный способ модуляции обеспечивает минимальную полосу частот группового сигнала. Подавление несущих достигается в индивидуальных модуляторах ($М_1$, $М_2$. ... $М_N$), которые, как правило, строятся по балансной схеме, а выделение одной боковой полосы (ОБП) осуществляется в полосовых фильтрах ($Ф_1$, $Ф_2$, ... $Ф_N$). Совокупность канальных сигналов на выходе суммирующего устройства Σ образует групповой сигнал. В групповом передатчике М групповой сигнал преобразуется в линейный сигнал, который и поступает в линию связи ЛС.

На приемной стороне линии связи линейный сигнал с помощью группового приемника П вновь преобразуется в групповой сигнал, из которого полосовыми фильтрами ($Ф_1$, $Ф_2$, ... $Ф_N$) выделяются канальные сигналы. После детектирования в канальных демодуляторах ($Д_1$, $Д_2$, ... $Д_N$) сигналы преобразуются в предназначенные получателям сообщения приемниками сообщений ($ПС_1$, $ПС_2$, ... $ПС_N$). Опорные колебания в канальных демодуляторах создаются с помощью генератора (ГНЧ). Сообщения, передаваемые по различным каналам, выделяются при помощи ФНЧ. .

Канальные передатчики вместе с суммирующим устройством образуют аппаратуру объединения. Групповой передатчик М, линия связи ЛС и групповой приемник П составляют групповой канал связи (тракт передачи), который вместе с аппаратурой объединения и индивидуальными приемниками составляет систему многоканальной связи. В составе технических устройств на передающей стороне многоканальной системы должна быть предусмотрена аппаратура объединения, а на приемной стороне - аппаратура разделения.

В общем случае групповой сигнал может формироваться не только простейшим суммированием канальных сигналов, но также и определенной логической обработкой, в результате которой каждый элемент группового сигнала несет информацию о сообщениях источников. Это так называемые системы с комбинационным разделением.

Чтобы разделяющие устройства были в состоянии различать сигналы отдельных каналов, должны существовать определенные признаки, присущие только данному сигналу. Такими признаками в общем случае могут быть параметры переносчика, например амплитуда, частота или фаза в случае непрерывной модуляции гармонического переносчика. При дискретных видах модуляции различающим признаком может служить и форма сигналов. Соответственно различаются и способы разделения сигналов: частотный, временной, фазовый и др.

Поскольку всякая реальная линия связи обладает ограниченной полосой пропускания, то при многоканальной передаче каждому отдельному каналу отводится определенная часть общей полосы пропускания.

На приемной стороне одновременно действуют сигналы всех каналов, различающиеся положением их частотных спектров на шкале частот. Чтобы без взаимных помех разделить такие сигналы, приемные устройства должны содержать частотные фильтры. Каждый из фильтров должен пропустить без ослабления лишь те частоты, которые принадлежат сигналу данного канала; частоты сигналов всех других каналов фильтр должен подавить.

Для снижения переходных помех до допустимого уровня приходится вводить защитные частотные интервалы (Рис. 11)

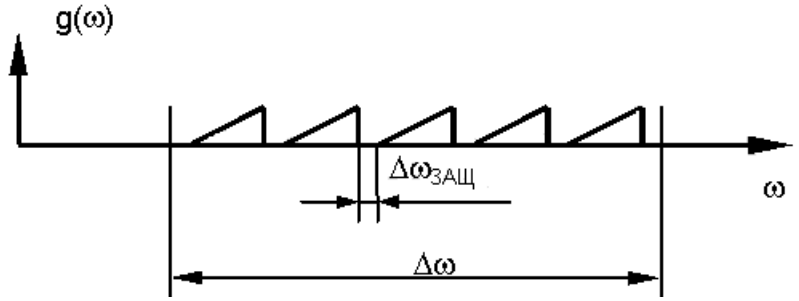


Рисунок 11. Спектр группового сигнала с защитными интервалами

При однократном преобразовании спектра сигнала (***fн = 104 кГц***) необходим специальный фильтр. При двукратном преобразовании спектра в первом модуляторе выбирается несущая до 30 кГц, например, ***f1 = 12 кГц***. При этом фильтр легко реализуется на ***LC***-элементах. На второй модулятор сигнал подается уже в полосе частот ***12,3…15,4 кГц***, и для переноса этого сигнала в заданную полосу частот необходимо использовать несущую ***f2 = 104 - f1 = 92 кГц***. Фильтр второго преобразователя частоты также легко реализуется на ***LC***-элементах.

Методы построения многоканальной аппаратуры с ЧРК отличаются способом формирования группового сигнала и особенностями передачи его в линейном тракте. По способу формирования группового сигнала (первый признак) различают:
1. метод с индивидуальным преобразованием сигналов;
2. метод с групповым преобразованием сигналов.
По способу усиления группового (линейного) сигнала (второй признак) выделяют:
1. метод с усилением каждого индивидуального сигнала;
2. метод с усилением линейного сигнала в целом.

При индивидуальном преобразовании сигналов формирование группового (линейного) спектра частот производится путем отдельного независимого преобразования каждого из ***N*** сигналов. Другими словами, при индивидуальном методе преобразователи, фильтры, усилители и другие элементы для каждого канала являются отдельными и повторяются в составе оконечной промежуточной аппаратуры столько раз, на сколько каналов рассчитана система передачи. Индивидуальные методы преобразования в оконечных и усиления в промежуточных станциях поясняются на рис. 12.

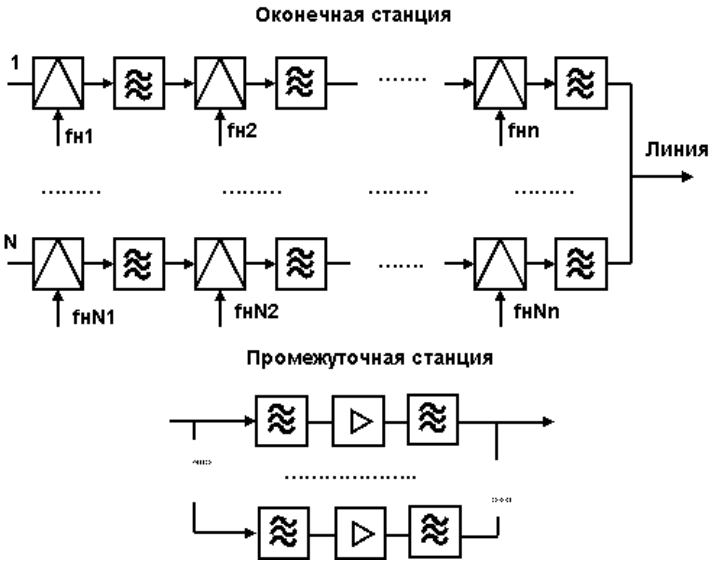


Рисунок 12. Индивидуальный метод преобразования

В основу метода группового преобразования сигналов положен принцип формирования линейного спектра в оконечном пункте МКС с помощью нескольких ступеней преобразования. На каждой ступени объединяются несколько канальных сигналов, т.е. линейный сигнал представляет собой сумму нескольких промежуточных групповых сигналов. При этом, в отличие от индивидуального метода, отдельной для каждого канала является только часть аппаратуры, а остальное оборудование – общее для всех каналов.

1.3 Лекция №3. Основные характеристики ЭВМ (2 часа)

1.3.1 Вопросы лекции:

1.Основные характеристики ЭВМ

1.3.2 Краткое содержание вопросов:

1. Основные характеристики ЭВМ

Первые электронные вычислительные машины (ЭВМ) появились немногим более полувека назад. За это время микроэлектроника, вычислительная техника и вся индустрия информатики стали одними из основных составляющих мирового научно-технического прогресса. Влияние вычислительной техники на все сферы деятельности человека продолжает распространяться вширь и вглубь. В настоящее время ЭВМ используются не только для выполнения сложных расчетов, но и в управлении производственными процессами, в образовании, здравоохранении, экологии и т.д. Это объясняется тем, что ЭВМ способны обрабатывать любые виды информации: числовую, текстовую, табличную, графическую, видео, звуковую.

Электронная вычислительная машина — это комплекс технических и программных средств, предназначенный для автоматизации подготовки и решения задач пользователей. Под пользователем понимают человека, в интересах которого проводится обработка данных на ЭВМ. В качестве пользователя могут выступать заказчики вычислительных работ, программисты, операторы. Как правило, время подготовки задач во много раз превышает время их решения.

Требования пользователей к выполнению вычислительных работ удовлетворяются специальным подбором и настройкой технических и программных средств. Обычно эти средства взаимосвязаны и объединяются в одну структуру.

Структура — совокупность элементов и их связей. Различают структуры технических, программных и аппаратно-программных средств. Выбирая ЭВМ для решения своих задач, пользователь интересуется функциональными возможностями технических и программных

модулей (как быстро может быть решена задача, насколько ЭВМ подходит для решения данного круга задач, какой сервис программ имеется в ЭВМ, возможности диалогового режима, стоимость подготовки и решения задач и т.д.). При этом пользователь интересуется не конкретной технической и программной реализацией отдельных модулей, а общими вопросами организации вычислений. Последнее включается в понятие архитектуры ЭВМ, содержание которого достаточно обширно.

Архитектура ЭВМ — это многоуровневая иерархия аппаратно-программных средств, из которых строится ЭВМ. Каждый из уровней допускает многовариантное построение и применение. Конкретная реализация уровней определяет особенности структурного построения ЭВМ. В последующих разделах учебника эти вопросы подробно рассматриваются.

Детализацией архитектурного и структурного построения ЭВМ занимаются различные категории специалистов вычислительной техники. Инженеры-схемотехники проектируют отдельные технические устройства и разрабатывают методы их сопряжения друг с другом. Системные программисты создают программы управления техническими средствами, информационного взаимодействия между уровнями, организации вычислительного процесса. Программисты-прикладники разрабатывают пакеты программ более высокого уровня, которые обеспечивают взаимодействие пользователей с ЭВМ и необходимый сервис при решении ими своих задач. Перечисленные специалисты рассматривают понятие архитектуры в более узком смысле. Для них наиболее важные структурные особенности сосредоточены в наборе команд ЭВМ, разграничивающем аппаратные и программные средства.

Сами же пользователи ЭВМ, которые обычно не являются профессионалами в области вычислительной техники, рассматривают архитектуру через более высокоуровневые аспекты, касающиеся их взаимодействия с ЭВМ (человеко-машинного интерфейса), начиная со следующих групп характеристик ЭВМ, определяющих ее структуру:

• технические и эксплуатационные характеристики ЭВМ (быстродействие и производительность, показатели надежности, достоверности, точности, емкость оперативной и внешней памяти, габаритные размеры, стоимость технических и программных средств, особенности эксплуатации и др.);

• характеристики и состав функциональных модулей базовой конфигурации ЭВМ; возможность расширения состава технических и программных средств; возможность изменения структуры;

• состав программного обеспечения ЭВМ и сервисных услуг (операционная система или среда, пакеты прикладных программ, средства автоматизации программирования).

Важнейшими характеристиками ЭВМ являются быстродействие и производительность. И хотя эти характеристики тесно связаны, тем не менее их не следует смешивать. Быстродействие, характеризуется

• стандартные универсальные тесты для ЭВМ, предназначенных для крупномасштабных вычислений (например, пакет математических задач Linpack, по которому ведется список TOP 500, включающий 500 самых производительных компьютерных установок в мире);

• специализированные тесты для конкретных областей применения компьютеров (например, для тестирования ПК по критериям офисной группы приложений используется тест Winstone97-Business, для группы «домашних компьютеров» — WinBench97-CPUMark32, а для группы ПК для профессиональной работы — 3DWinBench97-UserScene).

Отметим, что результаты оценивания ЭВМ по различным тестам несопоставимы. Наборы тестов и области применения компьютеров должны быть адекватны.

Другой важнейшей характеристикой ЭВМ является емкость запоминающих устройств. Она измеряется количеством структурных единиц информации, которые одновременно можно разместить в памяти. Этот показатель позволяет определить, какой набор программ и данных может быть одновременно размещен в памяти.

Наименьшей структурной единицей информации является бит — одна двоичная цифра. Как правило, емкость памяти оценивается в более крупных единицах измерения — байтах

(байт равен восьми битам). Следующими единицами измерения служат: 1Кбайт=210 байта=1024 байта, 1Мбайт =210 Кбайта=220 байта, 1 Гбайт=210 Мбай-та=220 Кбайта=230 байта.

Обычно отдельно характеризуют емкость оперативной памяти и емкость внешней памяти. Современные персональные ЭВМ могут иметь емкость оперативной памяти, равную 64 — 256 Мбайтам и даже больше. Этот показатель очень важен для определения, какие программные пакеты и их приложения могут одновременно обрабатываться в машине.

Емкость внешней памяти зависит от типа носителя. Так, емкость одной дискеты составляет 1,2; 1,4; 2,88 Мбайта в зависимости от типа дисковода и характеристик дискет. Емкость жесткого диска и дисков DVD может достигать нескольких десятков Гбайтов, емкость компакт-диска (CD-ROM) — сотни Мбайтов (640 Мбайт и выше) и т.д. Емкость внешней памяти характеризует объем программного обеспечения и отдельных программных продуктов, которые могут устанавливаться в ЭВМ. Например, для установки операционной среды Windows 2000 требуется объем памяти жесткого диска более 600 Мбайт и не менее 64 Мбайт оперативной памяти ЭВМ.

Надежность — это способность ЭВМ при определенных условиях выполнять требуемые функции в течение заданного времени (стандарт ISO (Международная организация стандартов) -2382/14-78).

числом определенного типа команд, выполняемых ЭВМ за одну секунду. Производительность — это объем работ (например, число стандартных программ), выполняемый ЭВМ в единицу времени.

Определение характеристик быстродействия и производительности представляет собой очень сложную инженерную и научную задачу, до настоящего времени не имеющую единых подходов и методов решения.

Казалось бы, что более быстродействующая вычислительная техника должна обеспечивать и более высокие показатели производительности. Однако практика измерений значений этих характеристик для разнотипных ЭВМ может давать противоречивые результаты. Основные трудности в решении данной задачи заключены в проблеме выбора: что и как измерять. Укажем лишь наиболее распространенные подходы.

Одной из альтернативных единиц измерения быстродействия была и остается величина, измеряемая в MIPS (Million Instructions Per Second — миллион операций в секунду). В качестве операций здесь обычно рассматриваются наиболее короткие операции типа сложения. MIPS широко использовалась для оценки больших машин второго и третьего поколений, но для оценки современных ЭВМ применяется достаточно редко по следующим причинам:

• набор команд современных микропроцессоров может включать сотни команд, сильно отличающихся друг от друга длительностью выполнения;

• значение, выраженное в MIPS, меняется в зависимости от особенностей программ;

• значение MIPS и значение производительности могут противоречить друг другу, когда оцениваются разнотипные вычислители (например, ЭВМ, содержащие сопроцессор для чисел с плавающей точкой и без такового).

При решении научно-технических задач в программах резко увеличивается удельный вес операций с плавающей точкой. Опять же для больших однопроцессорных машин в этом случае использовалась и продолжает использоваться характеристика быстродействия, выраженная в MFPOPS (Million Floating Point Operations Per Second — миллион операций с плавающей точкой в секунду). Для персональных ЭВМ этот показатель практически не применяется из-за особенностей решаемых задач и структурных характеристик ЭВМ.

Для более точных комплексных оценок существуют тестовые наборы, которые можно разделить на три группы:

• наборы тестов фирм-изготовителей для оценивания качества собственных изделий (например, компания Intel для своих микропроцессоров ввела показатель iCOMP-Intel Comparative Microprocessor Performance);

В аналоговых вычислительных машинах (АВМ) обрабатываемая информация представляется соответствующими значениями аналоговых величин: тока, напряжения, угла

поворота какого-то механизма и т.п. Эти машины обеспечивают приемлемое быстродействие, но

Высокая надежность ЭВМ закладывается в процессе ее производства. Переход на новую элементную базу — сверхбольшие интегральные схемы (СБИС) — резко сокращает число используемых интегральных схем, а значит, и число их соединений друг с другом. Хорошо продуманы компоновка компьютера и обеспечение требуемых режимов работы (охлаждение, защита от пыли). Модульный принцип построения позволяет легко проверять и контролировать работу всех устройств, проводить диагностику и устранять неисправности.

Точность — возможность различать почти равные значения (стандарт ISO — 2382/2-76). Точность получения результатов обработки в основном определяется разрядностью ЭВМ, которая в зависимости от класса ЭВМ может составлять 32, 64 и 128 двоичных разрядов.

Во многих применениях ЭВМ не требуется большой точности, например при обработке текстов и документов, при управлении технологическими процессами. В этом случае достаточно воспользоваться 8- и 16-разрядными двоичными кодами. При выполнении же сложных математических расчетов следует использовать высокую разрядность (32, 64 и даже более). Для работы с такими данными применяются соответствующие структурные единицы представления информации (байт, слово, двойное слово). Программными способами диапазон представления и обработки данных может быть увеличен в несколько раз, что позволяет достигать очень высокой точности.

Достоверность — свойство информации быть правильно воспринятой. Достоверность характеризуется вероятностью получения безошибочных результатов. Заданный уровень достоверности обеспечивается аппаратно-программными средствами контроля самой ЭВМ. Возможны методы контроля достоверности путем решения эталонных задач и повторных расчетов. В особо ответственных случаях проводятся контрольные решения на других ЭВМ и сравнение результатов.

1.4 Лекция №4. Минимальная конфигурация ЭВМ (2 часа)

1.4.1 Вопросы лекции:

1.Минимальная конфигурация ЭВМ

1.4.2 Краткое содержание вопросов:

1.Минимальная конфигурация ЭВМ

ЭВМ включает три основных устройства: системный блок, клавиатуру и дисплей (монитор). Однако для расширения функциональных возможностей ПЭВМ можно подключить различные периферийные устройства, в частности: печатающие устройства (принтеры), накопители на магнитной ленте (стримеры), различные манипуляторы (мышь, джойстик, трекбол, световое перо), устройства оптического считывания изображений (сканеры), графопостроители (плоттеры) и др.
Эти устройства подсоединяются к системному блоку с помощью кабелей через специальные гнезда (разъемы), которые размещаются обычно на задней стенке системного блока. В некоторых моделях ЭВМ при наличии свободных гнезд до-полнительные устройства вставляются непосредственно в системный блок, например, модем для обмена информацией с другими ПЭВМ через телефонную связь или стример для хранения больших массивов информации на МЛ.
Дисплей (монитор) - основное устройство для отображения информации, выводимой во время работы программ на ПЭВМ. Дисплеи могут существенно различаться, от их характеристик зависят возможности машин и используемого программного обеспечения. Различают дисплеи, пригодные для вывода лишь алфавитно-цифровой информации, и графические дисплеи. Другой важный признак - возможность поддержки цветного или только монохромного изображения. Важными техническими параметрами являются текстовой формат и разрешающая способность изображения. Текстовой формат (в текстовом режиме) характеризуется числом символов в строке и числом текстовых строк на экране. В графическом режи-

ме разрешающая способность задается числом точек по горизонтали и числом точечных строк по вертикали. Указанные параметры зависят как от конструкции экрана, так и от схемы управления, сосредоточенной в сис-темном блоке. В настоящее время в большинстве случаев применяется схема формирования изображения на основе растровой памяти (bit mapping). Каждый элемент изображения - одна точка на экране дисплея формируется из фрагмента растровой памяти, состоящего из 1, 2 или 4 бит. Информация, записанная в указанных битах, управляет яркостью (или цветом) точки на экране, а также ее миганием и другими возможными атрибутами.

Большинство профессиональных ПЭВМ использует дисплеи, основанные на цветных ЭЛТ (электронно-лучевых трубках). Наиболее часто в IBM-совместимых ПЭВМ используются мониторы типа SVGA.

В профессиональных ПЭВМ широко применяются цветные мониторы с очень высоким разрешением (1024x1024 и 2048x2048 точек) и возможностью получения изображений из 4096 базовых цветов, что обеспечивает до 16 млн. оттенков.

Общение пользователя с ПЭВМ облегчается с помощью различных манипуляторов. Наиболее распространенным из них является так называемая «мышь».

Мышь представляет собой устройство с двумя или тремя клавишами и утопленным свободно вращающимся в любом направлении шариком на нижней поверхности. Мышь подключается к компьютеру при помощи специального кабеля. Пользователь, перемещая мышь по поверхности стола (обычно для этого используются специальные резиновые коврики), позиционирует указатель мыши (стрелку, прямоугольник) на экране дисплея, а нажатием клавиш выполняет определенное действие, связанное с соответствующей клавишей (например, выполняет определенный пункт меню).

Мышь требует специальной программной поддержки.

В портативных ПЭВМ мышь обычно заменяется особым встроенным в клавиатуру шариком на подставке с двумя клавишами по бокам, называемым трекбол. Позиционирование указателя трекбола на экране дисплея производится вращением этого шарика. Клавиши трекбола имеют то же значение, что и клавиши мыши.

Для непосредственного считывания графической информации с бумажного или иного носителя в ПЭВМ применяются сканеры. Сканеры бывают настольные, позволяющие обрабатывать весь лист бумаги или пленки целиком, а также ручные. Ручные сканеры проводят над нужными рисунками или текстом, обеспечивая их считывание. Введенный при помощи сканера рисунок распознается ПЭВМ с помощью специального программного обеспечения. Рисунок может быть не только сохранён, но и откорректирован по желанию пользователя соответствующими графическими пакетами программ. В настоящее время выпускаются черно-белые и цветные сканеры с точностью разрешения до 8000 точек на дюйм (более 300 точек на 1 мм), однако эти устройства весьма дороги. Использование сканеров для непосредственного ввода в ПЭВМ текстовой информации с ее последующим редактированием затруднено также значительной сложностью программного обеспечения, необходимого для правильного распознавания и интерпретации отдельных символов.

К ручным манипуляторам относится и джойстик (joystick), пред-ставляющий собой подвижную рукоять с одной или двумя кнопками, при помощи которой можно позиционировать указатель на экране дисплея. Кнопки имеют то же назначение, что и клавиши мыши. Джойстик чаще используется в бытовых ПЭВМ, в первую очередь для игровых применений.

Таким образом, большинство современных ЭВМ строится на базе принципов, сформулированных американским учёным, одним из «отцов» кибернетики Дж. фон Нейманом. Эти принципы сводятся к следующему:

Основными блоками фон-неймановской машины являются блок управления, арифметико-логическое устройство, память и устройство ввода-вывода.

Информация кодируется в двоичной форме и разделяется на единицы, называемые словами.

Алгоритм представляется в форме последовательности управляющих слов, которые определяют смысл операции. Эти управляющие слова называются командами. Совокупность

команд, представляющая алгоритм, называется программой. Программы и данные хранятся в одной и той же памяти. Разнотипные слова различаются по способу использования, но не по способу кодирования. Устройство управления и арифметическое устройство обычно объединяются в одно, называемое центральным процессором. Они определяют действия, подлежащие выполнению, путем считывания команд из оперативной памяти. Обработка информации, предписанная алгоритмом, сводится к последовательному выполнению команд в порядке, однозначно определяемом программой.

1.5 Лекция №5 Состав команд и архитектура 8086 микропроцессора (2 часа)

1.5.1 Вопросы лекции:

1. Состав команд и архитектура 8086 микропроцессора

1.5.2 Краткое содержание вопросов:

1. Состав команд и архитектура 8086 микропроцессора

Архитектура ЭВМ — это абстрактное представление ЭВМ, которое отражает ее структурную, схемотехническую и логическую организацию. Понятие архитектуры ЭВМ включает в себя:

-структурную схему ЭВМ;

-средства и способы доступа к элементам структурной схемы ЭВМ; -организацию и разрядность интерфейсов ЭВМ;

-набор и доступность регистров;

-организацию и способы адресации памяти;

-способы представления и форматы данных ЭВМ;

-набор машинных команд ЭВМ;

-форматы машинных команд;

-обработку нештатных ситуаций (прерываний).

Все современные ЭВМ обладают некоторыми общими и индивидуальными свойствами архитектуры. Индивидуальные свойства присущи только конкретной модели компьютера. Наличие общих архитектурных свойств обусловлено тем, что большинство типов существующих машин принадлежат 4 и 5-му поколениям ЭВМ фон-неймановской архитектуры. К числу общих архитектурных свойств и принципов можно отнести:

-Принцип хранимой программы. Согласно ему, код программы и ее данные находятся в одном адресном пространстве в оперативной памяти.

-Принцип микропрограммирования. Суть этого принципа заключается в том, что машинный язык не является той конечной субстанцией, которая физически приводит в действие процессы в машине. В состав процессора входит блок микропрограммного управления. Этот блок для каждой машинной команды имеет набор действий-сигналов, которые нужно сгенерировать для физического выполнения требуемой машинной команды.

-Линейное пространство памяти — совокупность ячеек памяти, которым последовательно присваиваются номера (адреса) О, 1, 2, ... Последовательное выполнение программ. Процессор выбирает из памяти команды строго последовательно. Для изменения прямолинейного хода выполнения программы или осуществления ветвления необходимо использовать специальные команды. Они называются командами условного и безусловного перехода.

-С точки зрения процессора нет принципиальной разницы между данными и командами. Данные и машинные команды находятся в одном пространстве памяти в виде последовательности нулей и единиц. Это свойство связано с предыдущим. Процессор, исполняя содержимое некоторых последовательных ячеек памяти, всегда пытается трактовать его как коды машинной команды, а если это не так, то происходит аварийное завершение программы, содержащей некорректный фрагмент. Поэтому важно в программе всегда четко разделять пространство данных и команд.

-Безразличие к целевому назначению данных. Машине все равно, какую логическую нагрузку несут обрабатываемые ею данные.

Набор регистров.

Программная модель микропроцессора содержит 32 регистра, в той или иной мере доступных для использования программистом. Их можно разделить на две большие группы:

-16 пользовательских регистров;

- 16 системных регистров.

Пользовательские регистры

Пользовательскими регистрами программист может пользоваться и при написании своих программ. К этим регистрам относятся:

-восемь 32-битных регистров, которые могут использоваться программистами для хранения данных и адресов (их еще называют регистрами общего назначения (РОН)): eax/ax/ah/al, ebx/bx/bh/bl, edx/dx/dh/dl, ecx/cx/ch/cl, ebp/bp, esi/si, edi/di, esp/sp

-шесть регистров сегментов: cs, ds, ss, es, fs, gs;

-регистры состояния и управления: регистр флагов eflags/flags и регистр указателя команды eip/ip.

Регистры общего назначения

Все регистры этой группы позволяют обращаться к своим «младшим» частям; использовать для самостоятельной адресации можно только младшие 16- и 8-битные части этих регистров. Старшие 16 бит этих регистров как самостоятельные объекты недоступны. Так как регистры общего назначения находятся в процессоре внутри арифметико-логического устройства (АЛУ), то их еще называют регистрами АЛУ:

- eax/ax/ah/al (Accumulator register) — аккумулятор. Применяется для хранения промежуточных данных. В некоторых командах использование этого регистра обязательно;

- ebx/bx/bh/bl (Base register) — базовый регистр. Применяется для хранения базового адреса некоторого объекта в памяти;

- ecx/cx/ch/cl (Count register) — регистр-счетчик. Применяется в командах, производящих некоторые повторяющиеся действия. Его использование зачастую неявно и скрыто в алгоритме работы соответствующей команды. Например, команда организации цикла loop кроме передачи управления команде, находящейся по некоторому адресу, анализирует и уменьшает на единицу значение регистра ecx/cx;

- edx/dx/dh/dl (Data register) — регистр данных. Так же как и регистр eax/ax/ ah/al, он хранит промежуточные данные. В некоторых командах его использование обязательно; для некоторых команд это происходит неявно. Следующие два регистра используются для поддержки так называемых цепочечных операций, то есть операций, производящих последовательную обработку цепочек элементов, каждый из которых может иметь длину 32, 16 или 8 бит:

- esi/si (Source Index register) — индекс источника. Этот регистр в цепочечных операциях содержит текущий адрес элемента в цепочке-источнике;

- edi/di (Destination Index register) — индекс приемника (получателя). Этот регистр в цепочечных операциях содержит текущий адрес в цепочке-приемнике.

В архитектуре микропроцессора на программно-аппаратном уровне поддерживается такая структура данных, как стек. Для работы со стеком в системе команд микропроцессора есть специальные команды, а в программной модели микропроцессора для этого существуют специальные регистры:

- esp/sp (Stack Pointer register) — регистр указателя стека. Содержит указатель вершины стека в текущем сегменте стека;

- ebp/bp (Base Pointer register) — регистр указателя базы кадра стека. Предназначен для организации произвольного доступа к данным внутри стека.

Сегментные регистры

В программной модели микропроцессора имеется шесть сегментных регистров: cs, ss, ds, es, gs, fs. Их существование обусловлено спецификой организации и использования оперативной памяти микропроцессорами Intel. Она заключается в том, что микропроцессор аппаратно поддерживает структурную организацию программы в виде трех частей называемых сегментами.

Соответственно, такая организация памяти называется сегментной. Для того чтобы указать на сегменты, к которым программа имеет доступ в конкретный момент времени, предназначены сегментные регистры. Микропроцессор поддерживает следующие типы сегментов:

1. Сегмент кода. Содержит команды программы. Для доступа к этому сегменту служит регистр cs (code segment register) — сегментный регистр кода. Он содержит адрес сегмента с машинными командами, к которому имеет доступ микропроцессор (то есть эти команды загружаются в конвейер микропроцессора);

Сегмент данных. Содержит обрабатываемые программой данные. Для доступа к этому сегменту служит регистр ds (data segment register) — сегментный регистр данных, который хранит адрес сегмента данных текущей программы;

2. Сегмент стека. Этот сегмент представляет собой область памяти, называемую стеком. Работу со стеком микропроцессор организует по следующему принципу: последний записанный в эту область элемент выбирается первым. Для доступа к этому сегменту служит регистр ss (stack segment register) — сегментный регистр стека, содержащий адрес сегмента стека;

3. Дополнительный сегмент данных. Неявно алгоритмы выполнения большинства машинных команд предполагают, что обрабатываемые ими данные расположены в сегменте данных, адрес которого находится в сегментном регистре ds. Если программе недостаточно одного сегмента данных, то она имеет возможность использовать еще три дополнительных сегмента данных. Но в отличие от основного сегмента данных, адрес которого содержится в сегментном регистре ds, при использовании дополнительных сегментов данных их адреса требуется указывать явно с помощью специальных префиксов переопределения сегментов в команде. Адреса дополнительных сегментов данных должны содержаться в регистрах es, gs, fs (extension data segment/ registers).

4. Регистры состояния и управления

В микропроцессор включены несколько регистров, которые постоянно содержат информацию о состоянии как самого микропроцессора, так и программы, команды которой в данный момент загружены на конвейер. К этим регистрам относятся: 1- регистр флагов eflags/flags; 2- регистр указателя команды eip/ip.

Используя эти регистры, можно получать информацию о результатах выполнения команд и влиять на состояние самого микропроцессора. Назначение и содержимое этих регистров:

eflags/flags (flag register) — регистр флагов. Разрядность eflags/flags — 32/16 бит. Отдельные биты данного регистра имеют определенное функциональное назначение и называются флагами. Младшая часть этого регистра полностью аналогична регистру flags для i8086.

Флаги регистра eflags/flags можно разделить на три группы:

1- 8 флагов состояния. Эти флаги могут изменяться после выполнения машинных команд. Флаги состояния регистра eflags отражают особенности результата исполнения арифметических или логических операций. Это дает возможность анализировать состояние вычислительного процесса и реагировать на него с помощью команд условных переходов и вызовов подпрограмм.

2- 1 флаг управления. Обозначается как df (Directory Flag). Он находится в 10м бите регистра eflags и используется цепочечными командами.

3- 5 системных флагов, управляющих вводом/выводом, маскируемыми прерываниями, отладкой, переключением между задачами и виртуальным режимом 8086. Прикладным программам не рекомендуется модифицировать без необходимости эти флаги, так как в большинстве случаев это приведет к прерыванию работы программы.

Флаги состояния.

Мнемоника	Флаг	Номер бита в eflags	Содержание и назначение

флага			
cf	Флаг переноса(Car ry Flag)		1 — арифметическая операция произвела перенос из старшего бита результата. Старшим является 7-й, 15-й или 31-й бит в зависимости от размерности операнда; 0 — переноса не было
p^f	Флаг паритетами ty Flag)		1 — 8 младших разрядов (этот флаг — только для 8 младших разрядов операнда любого размера) результата содержат четное число единиц; 0 — 8 младших разрядов результата содержат нечетное число единиц
af	Вспомогател ь-ный флаг переноса (Auxiliary carry Flag)		Только для команд, работающих с BCD- числами. Фиксирует факт заема из младшей тетрады результата: 1 — в результате операции сложения был произведен перенос из разряда 3 в старший разряд или при вычитании был заем в разряд 3 младшей тетрады из значения в старшей тетраде; 0— переносов и заемов в (из) 3 разряд(а) младшей тетрады результата не было
zf	Флаг нуля(Zero Flag) Флаг знака(Sign Flag)	б	1 — результат нулевой; 0 — результат ненулевой
sf			Отражает состояние старшего бита результата (биты 7, 15 или 31 для 8, 16 или 32разрядных операндов соответственно):

			1— старший бит результата равен 1; 0— старший бит результата равен 0
of	Флаг переполнени я(Overflow Flag)		Флаг of используется для фиксирования факта потери значащего бита при арифметических операциях: 1— в результате операции происходит перенос (заем) в (из) старшего, знакового бита результата (биты 7, 15 или 31 для 8, 16 или 32разрядных операндов соответственно); 0 — в результате операции не происходит переноса (заема) в (из) старшего, знакового бита результата
iopl	Уровень привилегий ввода-вывода^^^ /Output Privilege Level)	12, 13	Используется в защищенном режиме работы микропроцессора, для контроля доступа к командам ввода-вывода в зависимости от привилегированности задачи
nt	Флаг вложенности задачи (Nested Task)		Используется в защищенном режиме работы микропроцессора для фиксации того факта, что одна задача вложена в другую

Системные флаги		
tf	Флаг трассировки (Trace Flag)	Предназначен для организации пошаговой работы микропроцессора: I — микропроцессор генерирует прерывание с номером 1 после выполнения каждой машинной команды. Может использоваться при отладке программ, в частности отладчиками; 0 — обычная работа
if	Флаг прерывания (Interrapt ena ble	Предназначен для разрешения или запрещения (маскирования) аппаратных прерываний (прерываний по входу INTR): I —

	Flag)	аппаратные прерывания разрешены; 0 — аппаратные прерывания запрещены
rf	Флаг возобновления (Resume Flag)	Используется при обработке прерываний от регистров отладки

vm	Флаг виртуального8086 (Virtual 8086 Mode)	Признак работы микропроцессора в режиме виртуального 8086: 1 — процессор работает в режиме виртуального 8086; 0 — процессор работает в реальном или защищенном режиме
ac	Флаг контроля выравнивания (Alignment Check)	Предназначен для разрешения контроля выравнивания при обращениях к памяти,И кратным 2 или 4, то установка данных битов приведет к тому, что все обращения по некратным адресам будут возбуждать исключительную ситуацию

eip/ip (Instruction Pointer register) — указатель команд. Регистр eip/ip имеет разрядность 32/16 бит и содержит смещение следующей подлежащей выполнению команды относительно содержимого сегментного регистра cs в текущем сегменте команд. Этот регистр непосредственно недоступен программисту, но загрузка и изменение его значения производятся различными командами управления, к которым относятся команды условных и безусловных переходов, вызова процедур и возврата из процедур. Возникновение прерываний также приводит к модификации регистра eip/ip.

1.6 Лекция №6. Сетевое оборудование (2 часа)

1.6.1 Вопросы лекции:

1. Сетевые адаптеры.
2. Концентраторы.
3. Коммутаторы.
4. Маршрутизатор

1.6.2 Краткое содержание вопросов:

1. Сетевые адаптеры.

Сетевой адаптер (Network Interface Card, NIC) - это периферийное устройство компьютера, непосредственно взаимодействующее со средой передачи данных, которая прямо или через другое коммуникационное оборудование связывает его с другими компьютерами. Это устройство решает задачи надежного обмена двоичными данными, представленными соответствующими электромагнитными сигналами, по внешним линиям связи. Как и любой контроллер компьютера, сетевой адаптер работает под управлением драйвера операционной системы и распределение функций между сетевым адаптером и драйвером может изменяться от реализации к реализации. Сетевые адаптеры и кабели являются аппаратной основой организации компьютерных сетей, их нормальная работа жизненно важна для сети. С кабелями и адаптерами связано обычно 80% неполадок в сети.

Сетевой адаптер вместе со своим драйвером реализует второй, канальный уровень модели открытых систем в конечном узле сети – компьютере.

Сетевой адаптер совместно с драйвером выполняет две операции: передачу и прием кадра.

В каждом компьютере должен быть установлен сетевой адаптер, обеспечивающий подключение к выбранному типу кабеля. Платы сетевого адаптера выступают в качестве физического интерфейса, или соединения между компьютером и сетевым кабелем. Платы вставляются в слоты расширения всех сетевых компьютеров и серверов.

В большинстве современных стандартов для локальных сетей предполагается, что между сетевыми адаптерами взаимодействующих компьютеров устанавливается специальное

коммуникационное устройство (концентратор, мост, коммутатор или маршрутизатор), которое берет на себя некоторые функции по управлению потоком данных.

Сетевой адаптер обычно выполняет следующие функции:

• Формирование передаваемой информации в виде кадра определенного формата. Кадр включает несколько служебных полей, среди которых имеется адрес компьютера назначения и контрольная сумма кадра, по которой сетевой адаптер станции назначения делает вывод о корректности доставленной по сети информации.

• Получение доступа к среде передачи данных. В локальных сетях в основном применяются разделяемые между группой компьютеров каналы связи (общая шина, кольцо), доступ к которым предоставляется по специальному алгоритму (наиболее часто применяются метод случайного доступа или метод с передачей маркера доступа по кольцу).

• Кодирование последовательности бит кадра последовательностью электрических сигналов при передаче данных и декодирование при их приеме.

• Преобразование информации из параллельной формы в последовательную и обратно. Эта операция связана с тем, что для упрощения проблемы синхронизации сигналов и удешевления линий связи в вычислительных сетях информация передается в последовательной форме, бит за битом, а не побайтно, как внутри компьютера.

• Синхронизация битов, байтов и кадров. Для устойчивого приема передаваемой информации необходимо поддержание постоянного синхронизма приемника и передатчика информации. Сетевой адаптер использует для решения этой задачи специальные методы кодирования, не использующие дополнительной шины с тактовыми синхросигналами. Эти методы обеспечивают периодическое изменение состояния передаваемого сигнала, которое используется тактовым генератором приемника для подстройки синхронизма. Кроме синхронизации на уровне битов, сетевой адаптер решает задачу синхронизации и на уровне байтов, и на уровне кадров.

Функцией сетевого адаптера является передача и прием сетевых сигналов из кабеля. Адаптер воспринимает команды и данные от сетевой операционной системы (ОС), преобразует эту информацию в один из стандартных форматов и передает ее в сеть через подключенный к адаптеру кабель.

Сетевые адаптеры различаются также по типу принятой в сети сетевой технологии - Ethernet, Token Ring, FDDI и т.п. Как правило, конкретная модель сетевого адаптера работает по определенной сетевой технологии (например, Ethernet). В связи с тем, что для каждой технологии сейчас имеется возможность использования различных сред передачи данных (тот же Ethernet поддерживает коаксиальный кабель, неэкранированную витую пару и оптоволоконный кабель), сетевой адаптер может поддерживать как одну, так и одновременно несколько сред. В случае, когда сетевой адаптер поддерживает только одну среду передачи данных, а необходимо использовать другую, применяются трансиверы и конверторы.

2. Концентраторы.

Концентратор (hub) – это сетевое устройство, предназначенное для объединения устройств сети в сегменты. Основной принцип его работы заключается в трансляции пакетов, поступающих на один из его портов на все другие порты. Таким образом, пакет, поступивший в сеть, будет отправлен всем остальным устройствам сети, т.е. будет осуществляться широковещательная передача. Концентратор работает на физическом уровне модели взаимодействия открытых систем (OSI). Концентратор используется в различных технологиях: ATM, xDSL, Token Ring, но наибольшее распространение он нашел в технологии Ethernet.

Концентратор можно рассматривать как репитер с несколькими выходами. В отличие от switch он не анализирует содержимое пакетов или их заголовки, а просто копирует их. Hub не позволяет увеличить число устройств в одном сегменте или разгрузить его, уменьшив число коллизий. Основная его задача – это подключение новых устройств к сети и организация ее топологии. Кроме того, hub может быть использован для организации резервных каналов.

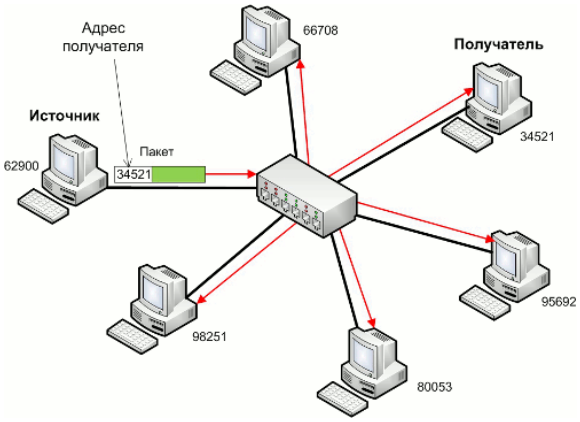


Рисунок 17. Пример работы сети с концентратором

Главным достоинством концентратора является простота реализации и, соответственно, невысокая стоимость. Однако из-за того, что он просто копирует пакеты во все свои порты, то в сети увеличивается вероятность возникновения коллизий. Это может привести к снижению скорости передачи и времени доставки пакетов. Именно поэтому вместо концентраторов обычно стараются применять коммутаторы, которые передают пакеты только к тому порту, к которому подключен компьютер получатель.

В зависимости от выполняемых задач можно встретить различные по емкости концентраторы от 4 до 64 портов. Однако это не предел. Они могут объединяться в более емкие устройства. Максимально возможное число работающих в спаренном режиме устройств ограничивается лишь характеристиками используемой технологии (для Ethernet – 1024 портов в одном сегменте). Концентраторы отличаются также по типу используемых проводников (витая пара, коаксиальный кабель) и используемой среде передачи (электрический или оптический кабель).

3. Коммутаторы.

Сетевой коммутатор (network switch) – это устройство, используемое в сетях передачи пакетов, предназначенное для объединения нескольких сегментов. В отличие от маршрутизатора (router) коммутатор работает на канальном уровне модели OSI, что и определяет главные различия между ними. Коммутатор не занимается расчетом маршрута для дальнейшей передачи пакетов по сети, анализируя различные факторы, как это делает маршрутизатор. Switch только передает данные от одного порта к другому на основе содержащейся в пакете информации. Обычно признаком выбора выходного порта служит MAC-адрес устройства, к которому передаются данные. В свою очередь коммутатор в отличие от концентратора или репитера не просто транслирует порты ко всем выходам, которые у него есть, а к одному, заранее выбранному.

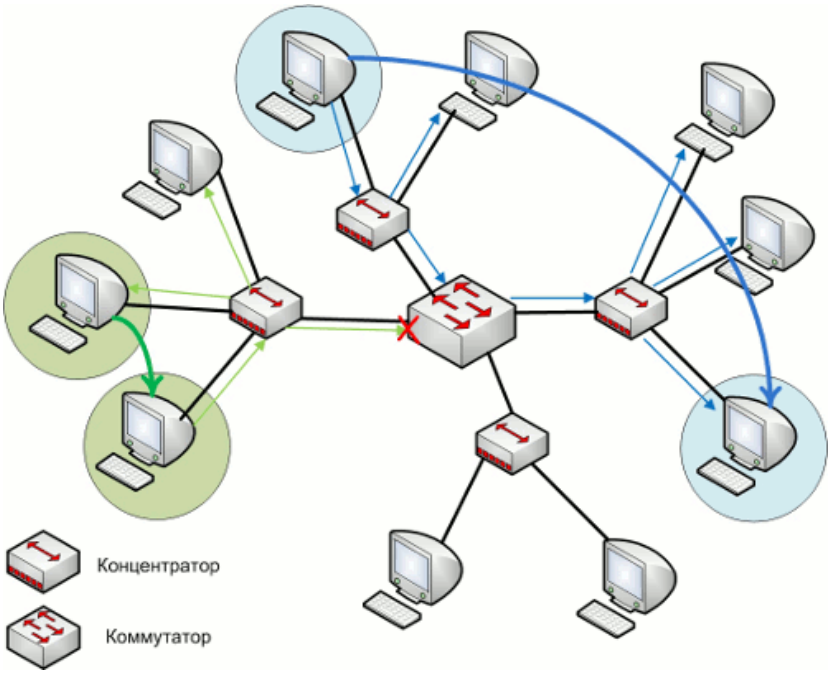


Рисунок 18. Пример сети с коммутатором

Сетевые коммутаторы применяются в нескольких технологиях, но наибольшее распространение нашли в Ethernet. Главной их задачей в сети Ethernet является разделение сети на сегменты. Это особенно актуально в сетях с большим числом рабочих станций, т.к. чем больше оконечных устройств работают одновременно с единой средой передачи данных тем выше вероятность возникновения коллизии (одновременной передачи данных несколькими устройствами) и, следовательно, ниже эффективность работы сети. Коммутатор позволяет разбить единую сеть на несколько сегментов и увеличить число одновременно работающих устройств.

Существую управляемые и неуправляемые коммутаторы. Неуправляемые коммутаторы самонастраиваются после включения в сеть. Они анализируют MAC-адреса всех устройств, подключенных к ним и будут осуществлять коммутацию между портами на основе анализа заголовка пакета, в котором содержится MAC-адресом устройства-получателя. Управляемые коммутаторы предоставляют интерфейс для администратора, который может выполнить его настройку для работы в конкретной сети. Например, есть возможность выбора режима защиты от отказа (в случае работы в паре с резервным коммутатором), объединения нескольких портов в единое направление, настройки приоритетов и резервирования портов и мн. др. Обычно управляемые коммутаторы дороже и используются в емких сетях, с дополнительными требованиями по надежности.

Switch может быть выполнен и в виде небольшой платы на 4 порта и многополочного штатива с возможность интеграции дополнительных устройств и расширения емкости. Также в зависимости от назначения сетевой коммутатор может снабжаться автономным питанием, портами управления и резервирования, охлаждением.

4. Маршрутизатор.

Маршрутизатор – это устройство пакетной сети передачи данных, предназначенное для объединения сегментов сети и ее элементов и служит для передачи пакетов между ними на основе каких-либо правил. Маршрутизаторы работают на сетевом (третьем) уровне модели OSI в качестве узловых устройств для различных технологий: IP, ATM, Frame Relay и мн. др.

Одной из самых важных задач маршрутизаторов является выбор оптимального маршрута передачи пакетов между подключенными сетями. Причем сделать это необходимо максимально оперативно с минимальной временной задержкой. Одновременно с этим должна отслеживаться текущая обстановка в сети для исключения из возможных путей доставки перегруженные и поврежденные участки. Практически все маршрутизаторы используют в

своей работе, так называемые, таблицы маршрутизации. Это своеобразные базы данных, которые содержат информацию обо всех возможных маршрутах передачи пакетов с некоторой дополнительной информацией, которая берется в расчет при выборе оптимального варианта доставки. Это может быть состояние канала, время доставки информации, загруженность, полоса пропускания и др.

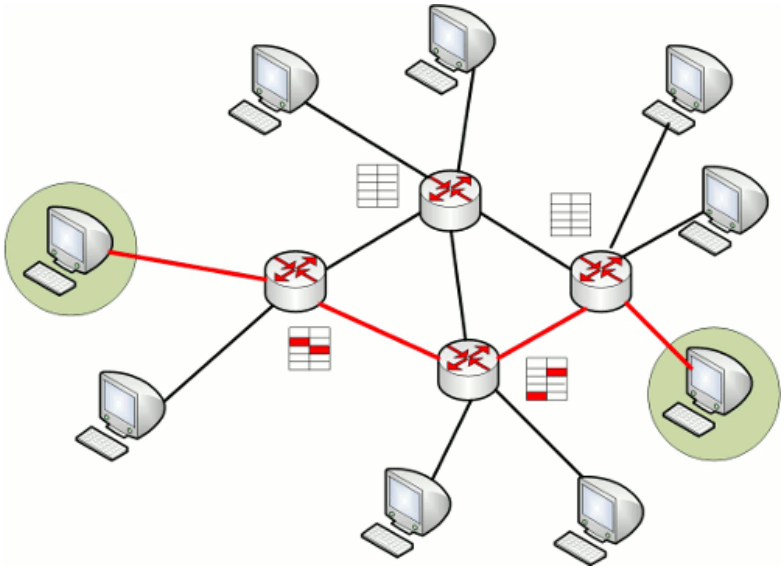


Рисунок 19. Пример работы маршрутизаторов в сети пакетной передачи данных

Важным аспектом работы маршрутизаторов является способ обновления информации в таблицах маршрутизации. Это может выполняться двумя способами вручную и автоматически. В первом случае администратор сети самостоятельно настраивает таблицы маршрутизации. Такой вариант подходит только для небольших сетей, конфигурация которых изменяется редко. Маршрутизаторы первого типа называются статическими. Автоматическое обновление таблиц маршрутизации выполняется с помощью обмена информационными сообщениями между соседними маршрутизаторами о текущей обстановке, а также проверке соединительных каналов между ними.Такие маршрутизаторы называются динамическими. Главный их недостаток заключается в необходимости дополнительных сетевых и вычислительных ресурсов для обмена данными и расчета маршрута. Однако динамические маршрутизаторы могут быть использованы при построении сетей любого масштаба.

Маршрутизаторы бывают как проводные – наиболее классический тип с несколькими портами, в которые подключаются кабели от внешних устройств, так и беспроводные, например, используемые для построения сетей WiFi. Также маршрутизаторы значительно различаются по емкости. Это могут быть как небольшие роутеры с 8-12 портами, которые используются при построении локальных сетей, так и громоздкие модульные конструкции, рассчитанные на сотни подключаемых сегментов.

1.7 Лекция №7. Протокол TCP/IP (2 часа)

1.7.1 Вопросы лекции:

1. Протоколы TCP/IP.
2. Архитектура TCP/IP.

1.7.2 Краткое содержание вопросов:

1. Протоколы TCP/IP.

TCP/IP - это два основных сетевых протокола Internet. Часто это название используют и для обозначения сетей, работающих на их основе. Протокол IP (Internet Protocol - IP v4) обеспечивает маршрутизацию (доставку по адресу) сетевых пакетов. Протокол TCP (Transfer Control Protocol) обеспечивает установление надежного соединения между двумя машинами и

собственно передачу данных, контролируя оптимальный размер пакета передаваемых данных и осуществляя перепосылку в случае сбоя. Число одновременно устанавливаемых соединений между абонентами сети не ограничивается, т. е. любая машина может в некоторый промежуток времени обмениваться данными с любым количеством других машин по одной физической линии.

Другое важное преимущество сети с протоколами TCP/IP состоит в том, что по нему могут быть объединены машины с разной архитектурой и разными операционными системами, например Unix, VAX VMS, MacOS, MS-DOS, MS Windows и т.д. Причем машины одной системы при помощи сетевой файловой системы NFS (Net File System) могут подключать к себе диски с файловой системой совсем другой ОС и оперировать "чужими" файлами как своими.

Протоколы TCP/IP (Transmission Control Protocol/Internet Protocol) являются базовыми транспортным и сетевым протоколами в OS UNIX. В заголовке TCP/IP пакета указывается:

IP-адрес отправителяIP-адрес получателяНомер порта (Фактически - номер прикладной программы, которой этот пакет предназначен)

Пакеты TCP/IP имеют уникальную особенность добраться до адресата, пройдя сквозь разнородные в том числе и локальные сети, используя разнообразные физические носители.Маршрутизацию IP-пакета (переброску его в требуемую сеть) осуществляют на добровольных началах компьютеры, входящие в TCP/IP сеть.

Протокол IP - это протокол, описывающий формат пакета данных, передаваемого по сети.

Следующий простой пример можетн прояснить, каким образом происходит передача данных и передача данных. Когда Вы получаете телеграмму, весь текст в ней (и адрес, и сообщение) написан на ленте подряд, но есть правила, позволяющие понять, где тут адрес, а где сообщение. Аналогично, пакет в компьютерной сети представляет собой поток битов, а протокол IP определяет, где адрес и прочая служебная информация, а где сами передаваемые данные. Таким образом, протокол IP в эталонной модели ISO/OSI является протоколомсетевого (3) уровня.

Протокол TCP - это протокол следующего уровня, предназначеный для контроля передачи и целостности передаваемой информации.

Когда Вы не расслышали, что сказал Вам собеседник в телефонном разговоре, Вы просите его повторить сказанное. Приблизительно этим занимается и протокол TCP применительно к компьютерным сетям. Компьютеры обмениваются пакетами протокола IP, контролируют их передачу по протоколу TCP и, объединяясь в глобальную сеть, образуют Интернет. Протокол TCP является протоколом транспортного (4) уровня.

2. Архитектура TCP/IP.

После того как семейство протоколов TCP/IP было реализовано и внедрено, Международная организация по стандартизации (International Organization for Standardization, ISO) предложила собственную семиуровневую сетевую модель, названную OSI (Open System Interconnection — взаимодействие открытых систем). Она так никогда и не приобрела широкой популярности из-за своей сложности и неэффективности.

В данной модели обмен информацией может быть представлен в виде стека, представленного на рисунке 21. Как видно из рисунка, в этой модели определяется все - от стандарта физического соединения сетей до протоколов обмена прикладного программного обеспечения. Дадим некоторые комментарии к этой модели.

Физический уровень данной модели определяет характеристики физической сети передачи данных, которая используется для межсетевого обмена. Это такие параметры, как: напряжение в сети, сила тока, число контактов на разъемах и т.п. Типичными стандартами этого уровня являются, например RS232C, V35, IEEE 802.3 и т.п.

Рисунок 21. Семиуровневая модель протоколов межсетевого обмена OSI

К канальному уровню отнесены протоколы, определяющие соединение, например, SLIP (Strial Line Internet Protocol), PPP (Point to Point Protocol), NDIS, пакетный протокол, ODI и т.п. В данном случае речь идет о протоколе взаимодействия между драйверами устройств и устройствами, с одной стороны, а с другой стороны, между операционной системой и драйверами устройства. Такое определение основывается на том, что драйвер - это, фактически, конвертор данных из оного формата в другой, но при этом он может иметь и свой внутренний формат данных.

К сетевому (межсетевому) уровню относятся протоколы, которые отвечают за отправку и получение данных, или, другими словами, за соединение отправителя и получателя. Вообще говоря, эта терминология пошла от сетей коммутации каналов, когда отправитель и получатель действительно соединяются на время работы каналом связи. Применительно к сетям TCP/IP, такая терминология не очень приемлема. К этому уровню в TCP/IP относят протокол IP (Internet Protocol). Именно здесь определяется отправитель и получатель, именно здесь находится необходимая информация для доставки пакета по сети.

Транспортный уровень отвечает за надежность доставки данных, и здесь, проверяя контрольные суммы, принимается решение о сборке сообщения в одно целое. В Internet транспортный уровень представлен двумя протоколами TCP (Transport Control Protocol) и UDP (User Datagramm Protocol). Если предыдущий уровень (сетевой) определяет только правила доставки информации, то транспортный уровень отвечает за целостность доставляемых данных.

Уровень сессии определяет стандарты взаимодействия между собой прикладного программного обеспечения. Это может быть некоторый промежуточный стандарт данных или правила обработки информации. Условно к этому уровню можно отнеси механизм портов протоколов TCP и UDP и Berkeley Sockets. Однако обычно, рамках архитектуры TCP/IP такого подразделения не делают.

Уровень обмена данными с прикладными программами (Presentation Layer) необходим для преобразования данных из промежуточного формата сессии в формат данных приложения. В Internet это преобразование возложено на прикладные программы.

Уровень прикладных программ или приложений определяет протоколы обмена данными этих прикладных программ. В Internet к этому уровню могут быть отнесены такие протоколы, как: FTP, TELNET, HTTP, GOPHER и т.п.

Вообще говоря, стек протоколов TCP отличается от только что рассмотренного стека модели OSI. Обычно его можно представить в виде схемы, представленной на рисунке 22.

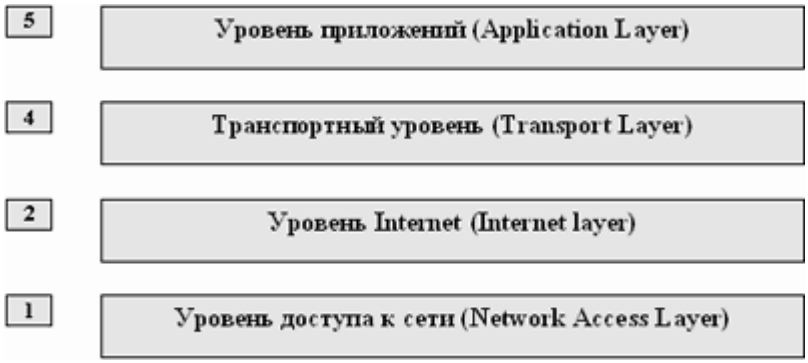


Рисунок 22. Структура стека протоколов TCP/IP

В этой схеме на уровне доступа к сети располагаются все протоколы доступа к физическим устройствам. Выше располагаются протоколы межсетевого обмена IP, ARP, ICMP. Еще выше основные транспортные протоколы TCP и UDP, которые кроме сбора пакетов в сообщения еще и определяют какому приложению необходимо данные отправить или от какого приложения необходимо данные принять. Над транспортным уровнем располагаются протоколы прикладного уровня, которые используются приложениями для обмена данными.

Базируясь на классификации OSI (Open System Integration) всю архитектуру протоколов семейства TCP/IP попробуем сопоставить с эталонной моделью (рисунок 23).

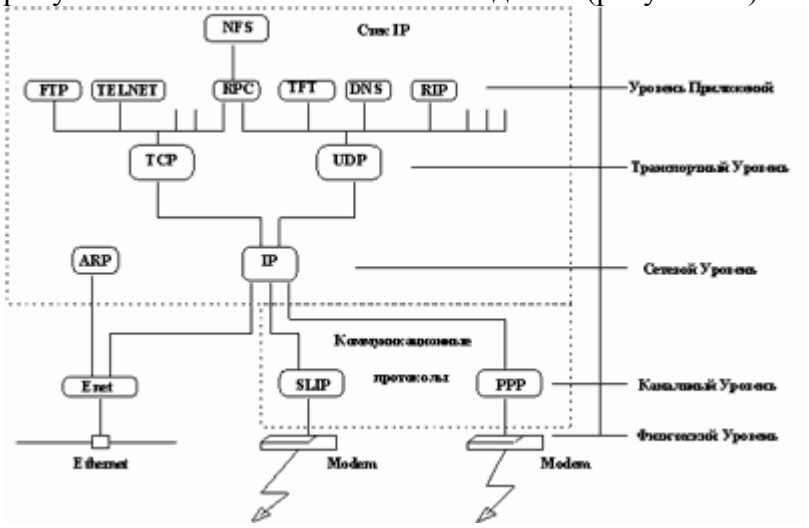


Рисунок 23. Схема модулей, реализующих протоколы семейства TCP/IP в узле сети

Прямоугольниками на схеме обозначены модули, обрабатывающие пакеты, линиями - пути передачи данных. Прежде чем обсуждать эту схему, введем необходимую для этого терминологию.

Драйвер - программа, непосредственно взаимодействующая с сетевым адаптером.

Модуль - это программа, взаимодействующая с драйвером, с сетевыми прикладными программами или с другими модулями.

Схема приведена для случая подключения узла сети через локальную сеть Ethernet, поэтому названия блоков данных будут отражать эту специфику.

Сетевой интерфейс - физическое устройство, подключающее компьютер к сети. В нашем случае - карта Ethernet.

Кадр - это блок данных, который принимает/отправляет сетевой интерфейс.

IP-пакет - это блок данных, которым обменивается модуль IP с сетевым интерфейсом.

UDP-датаграмма - блок данных, которым обменивается модуль IP с модулем UDP.

TCP-сегмент - блок данных, которым обменивается модуль IP с модулем TCP.

Прикладное сообщение - блок данных, которым обмениваются программы сетевых приложений с протоколами транспортного уровня.

Инкапсуляция - способ упаковки данных в формате одного протокола в формат другого протокола. Например, упаковка IP-пакета в кадр Ethernet или TCP-сегмента в IP-пакет. Согласно словарю иностранных слов термин "инкапсуляция" означает "образование капсулы вокруг чужих для организма веществ (инородных тел, паразитов и т.д.)". В рамках межсетевого обмена понятие инкапсуляции имеет несколько более расширенный смысл. Если в случае инкапсуляции IP в Ethernet речь идет действительно о помещении пакета IP в качестве данных Ethernet-фрейма, или, в случае инкапсуляции TCP в IP, помещение TCP-сегмента в качестве данных в IP-пакет, то при передаче данных по коммутируемым каналам происходит дальнейшая "нарезка" пакетов теперь уже на пакеты SLIP или фреймы PPP.

Рисунок 24. Инкапсуляция протоколов верхнего уровня в протоколы TCP/IP

Вся схема (рисунок 24) называется стеком протоколов TCP/IP или просто стеком TCP/IP. Чтобы не возвращаться к названиям протоколов расшифруем аббревиатуры TCP, UDP, ARP, SLIP, PPP, FTP, TELNET, RPC, TFTP, DNS, RIP, NFS:

TCP - Transmission Control Protocol - базовый транспортный протокол, давший название всему семейству протоколов TCP/IP.

UDP - User Datagram Protocol - второй транспортный протокол семейства TCP/IP. Различия между TCP и UDP будут обсуждены позже.

ARP - Address Resolution Protocol - протокол используется для определения соответствия IP-адресов и Ethernet-адресов.

SLIP - Serial Line Internet Protocol (Протокол передачи данных по телефонным линиям).

PPP - Point to Point Protocol (Протокол обмена данными "точка-точка").

FTP - File Transfer Protocol (Протоколобменафайлами).

TELNET - протокол эмуляции виртуального терминала.

RPC - Remote Process Control (Протокол управления удаленными процессами).

TFTP - Trivial File Transfer Protocol (Тривиальный протокол передачи файлов).

DNS - Domain Name System (Системадоменныхимен).

RIP - Routing Information Protocol (Протоколмаршрутизации).

NFS - Network File System (Распределенная файловая система и система сетевой печати).

При работе с такими программами прикладного уровня, как FTP или telnet, образуется стек протоколов с использованием модуля TCP, представленный на рисунке 25.

Рисунок 25. Стек протоколов при использовании модуля TCP

При работе с прикладными программами, использующими транспортный протокол UDP, например, программные средства Network File System (NFS), используется другой стек, где вместо модуля TCP будет использоваться модуль UDP (рисунок 26).

Рисунок 26. Стек протоколов при работе через транспортный протокол UDP

При обслуживании блочных потоков данных модули TCP, UDP и драйвер ENET работают как мультиплексоры, т.е. перенаправляют данные с одного входа на несколько выходов и наоборот, с многих входов на один выход. Так, драйвер ENET может направить кадр либо модулю IP, либо модулю ARP, в зависимости от значения поля "тип" в заголовке кадра. Модуль IP может направить IP-пакет либо модулю TCP, либо модулю UDP, что определяется полем "протокол" в заголовке пакета.

Получатель UDP-датаграммы или TCP-сообщения определяется на основании значения поля "порт" в заголовке датаграммы или сообщения.

Все указанные выше значения прописываются в заголовке сообщения модулями на отправляющем компьютере. Так как схема протоколов - это дерево, то к его корню ведет только один путь, при прохождении которого каждый модуль добавляет свои данные в заголовок блока. Машина, принявшая пакет, осуществляет демультиплексирование в соответствии с этими отметками.

Технология Internet поддерживает разные физические среды, из которых самой распространенной является Ethernet. В последнее время большой интерес вызывает подключение отдельных машин к сети через TCP-стек по коммутируемым (телефонным) каналам. С появлением новых магистральных технологий типа ATM или FrameRelay активно ведутся исследования по инкапсуляции TCP/IP в эти протоколы. На сегодняшний день многие проблемы решены и существует оборудование для организации TCP/IP сетей через эти системы.

1.8 Лекция №8.-9 Типовая структура процессора (4 часа)

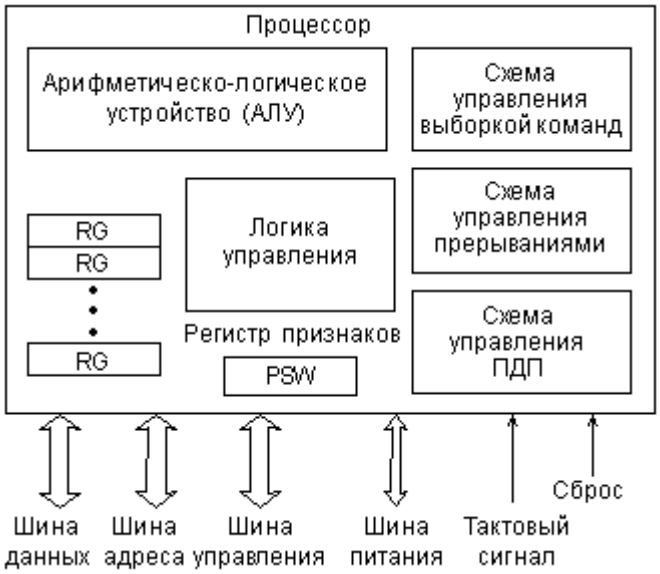
1.8.1 Вопросы лекции:

1. Типовая структура процессора

1.8.2 Краткое содержание вопросов:

1. Типовая структура процессора

Упрощенно структуру микропроцессора можно представить в следующем виде:



Внутренняя структура микропроцессора.

Основные функции показанных узлов следующие:

Схема управления выборкой команд выполняет чтение команд из памяти и их дешифрацию. В первых микропроцессорах было невозможно одновременное выполнение предыдущей команды и выборка следующей команды, так как процессор не мог совмещать эти операции. Но уже в 16-разрядных процессорах появляется так называемый *конвейер* (очередь) команд, позволяющий выбирать несколько следующих команд, пока выполняется предыдущая. Два процесса идут параллельно, что ускоряет работу процессора. Конвейер представляет собой небольшую внутреннюю память процессора, в которую при малейшей возможности (при освобождении внешней шины) записывается несколько команд, следующих за исполняемой. Читаются эти команды процессором в том же порядке, что и записываются в конвейер (это память типа **FIFO**, First In — First Out, первый вошел — первый вышел). Правда, если выполняемая команда предполагает переход не на следующую ячейку памяти, а на удаленную (с меньшим или большим адресом), конвейер не помогает, и его приходится сбрасывать. Но такие команды встречаются в программах сравнительно редко.

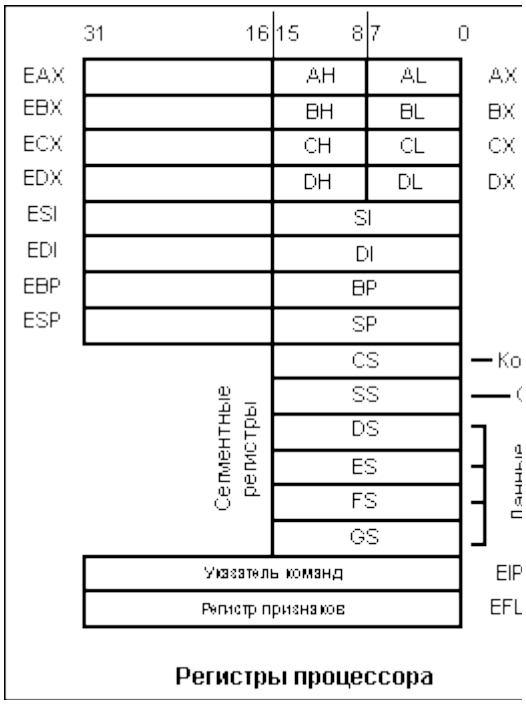
Развитием идеи конвейера стало использование внутренней кэш-памяти процессора, которая заполняется командами, пока процессор занят выполнением предыдущих команд. Чем больше объем кэш-памяти, тем меньше вероятность того, что ее содержимое придется сбросить при команде перехода. Понятно, что обрабатывать команды, находящиеся во внутренней памяти, процессор может гораздо быстрее, чем те, которые расположены во внешней памяти. В кэш-памяти могут храниться и данные, которые обрабатываются в данный момент, это также ускоряет работу. Для большего ускорения выборки команд в современных процессорах применяют совмещение выборки и дешифрации, одновременную дешифрацию нескольких команд, несколько параллельных конвейеров команд, предсказание команд переходов и некоторые другие методы.

Арифметико-логическое устройство (или **АЛУ**, ALU) предназначено для обработки информации в соответствии с полученной процессором командой. Примерами обработки могут служить логические операции (типа логического «И», «ИЛИ», «Исключающего ИЛИ» и т.д.) то есть побитные операции над операндами, а также арифметические операции (типа сложения, вычитания, умножения, деления и т.д.). Над какими кодами производится операция,

куда помещается ее результат — определяется выполняемой командой. Если команда сводится всего лишь к пересылке данных без их обработки, то АЛУ не участвует в ее выполнении.

Быстродействие АЛУ во многом определяет производительность процессора. Причем важна не только частота тактового сигнала, которым тактируется АЛУ, но и количество тактов, необходимое для выполнения той или иной команды. Для повышения производительности разработчики стремятся довести время выполнения команды до одного такта, а также обеспечить работу АЛУ на возможно более высокой частоте. Один из путей решения этой задачи состоит в уменьшении количества выполняемых АЛУ команд, создание процессоров с уменьшенным набором команд (так называемые RISC-процессоры). Другой путь повышения производительности процессора — использование нескольких параллельно работающих АЛУ.

Что касается операций над числами с плавающей точкой и других специальных сложных операций, то в системах на базе первых процессоров их реализовали последовательностью более простых команд, специальными подпрограммами, однако затем были разработаны специальные вычислители — математические сопроцессоры, которые заменяли основной процессор на время выполнения таких команд. В современных микропроцессорах математические сопроцессоры входят в структуру как составная часть.



Регистры процессора

Регистры процессора представляют собой по сути ячейки очень быстрой памяти и служат для временного хранения различных кодов: данных, адресов, служебных кодов. Операции с этими кодами выполняются предельно быстро, поэтому, в общем случае, чем больше внутренних регистров, тем лучше. Кроме того, на быстродействие процессора сильно влияет разрядность регистров. Именно разрядность регистров и АЛУ называется *внутренней разрядностью процессора*, которая может не совпадать с внешней разрядностью.

По отношению к назначению внутренних регистров существует два основных подхода. Первого придерживается, например, компания Intel, которая каждому регистру отводит строго определенную функцию. С одной стороны, это упрощает организацию процессора и уменьшает время выполнения команды, но с другой — снижает гибкость, а иногда и замедляет работу программы. Например, некоторые арифметические операции и обмен с устройствами ввода/вывода проводятся только через один регистр — *аккумулятор*, в результате чего при выполнении некоторых процедур может потребоваться несколько дополнительных пересылок между регистрами. Второй подход состоит в том, чтобы все (или почти все) регистры сделать равноправными, как , например, в 16-разрядных процессорах Т-11 фирмы DEC. При этом достигается высокая гибкость, но необходимо усложнение структуры процессора. Существуют и промежуточные решения, в частности, в процессоре MC68000 фирмы Motorola половина регистров использовалась для данных, и они были взаимозаменяемы, а другая половина — для адресов, и они также взаимозаменяемы.

В первую группу входят *регистры общего назначения*. В процессорах 386 и выше имеются восемь 32-битовых регистров общего назначения **EAX, EBX, ECX, EDX, ESI, EDI, EBP**, и **ESP**. Процессоры 386 и выше могут обращаться к 16-битовым половинам 32-битовых регистров. При необходимости возможна работа с половинами регистров, поскольку они разделены на старшую и младшую половину, называемые **AH** и **AL, BH** и **BL** и т.д. Такое разделение регистров имеется во всех

процессорах. Значительная часть внутренних операций компьютеров производится с использованием регистров общего назначения.

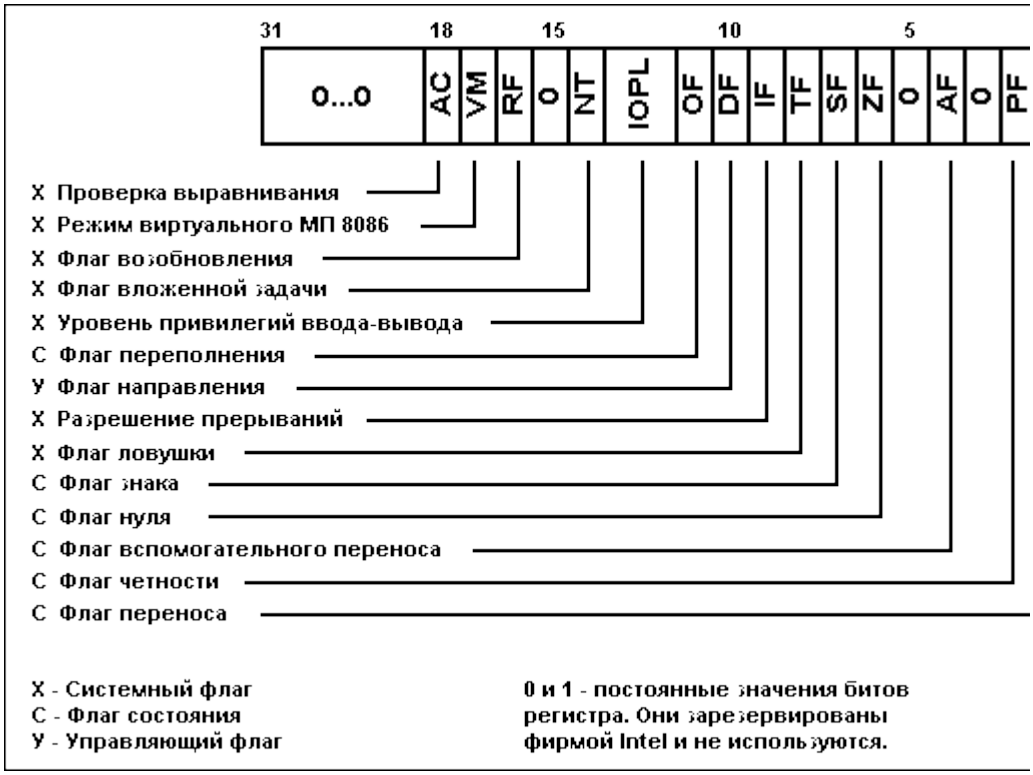
Следующая группа из шести регистров помогает процессору обращаться к памяти. Они называются *сегментными регистрами* и каждый из них помогает обращаться к области (или сегменту) памяти. В прежних процессорах размер сегментов составлял 64 Кбайт, а в новых процессорах длина сегмента переменная и варьируется от одного байта до 4 Гбайт.

Регистр **CS** сегмента кода (программы) показывает, в каком месте памяти находится программа. Регистр **DS** сегмента данных локализует используемые программой данные. Регистр **ES** дополнительного сегмента дополняет сегмент данных. Регистр **SS** сегмента стека определяет стек компьютера. В процессорах 386 и выше имеются еще два сегментных регистра: **FS** и **GS**, предназначенных для адресации памяти.

Если сегментные регистры обеспечивают доступ к большим блокам памяти, то последняя группа используется совместно с сегментным регистром для локализации в памяти конкретных байтов. *Регистр указателя команды* **IP** определяет ту точку, где выполняется программа. Регистры *указателя стека* **SP** и*указателя базы* **BP** помогают следить за информацией в стеке (стек — это область памяти, где хранится информация о текущих действиях компьютера). Регистры *индекса источника* **SI** и *индекса получателя* **DI** помогают программам пересылать большие блоки данных из одного места в другое.

Регистр признаков (регистр состояния) занимает особое место, хотя он также является внутренним регистром процессора. Содержащаяся в нем информация — это не данные, не адрес, а слово состояния процессора (ССП, **PSW** — Processor Status Word).

Каждый бит этого слова (флаг) содержит информацию о результате предыдущей команды. Например, есть бит нулевого результата, который устанавливается в том случае, когда результат выполнения предыдущей команды — нуль, и очищается в том случае, когда результат выполнения команды отличен от нуля. Эти биты (флаги) используются командами условных переходов, например, командой перехода в случае нулевого результата. В этом же регистре иногда содержатся флаги управления, определяющие режим выполнения некоторых команд.



Назначение битов регистра флагов

Схема управления прерываниями обрабатывает поступающий на процессор запрос прерывания, определяет адрес начала программы обработки прерывания (адрес вектора прерывания), обеспечивает переход к этой программе после выполнения текущей команды и

сохранения в памяти (в стеке) текущего состояния регистров процессора. По окончании программы обработки прерывания процессор возвращается к прерванной программе с восстановленными из памяти (из стека) значениями внутренних регистров.

Схема управления прямым доступом к памяти служит для временного отключения процессора от внешних шин и приостановки работы процессора на время предоставления прямого доступа запросившему его устройству.

Логика управления организует взаимодействие всех узлов процессора, перенаправляет данные, синхронизирует работу процессора с внешними сигналами, а также реализует процедуры ввода и вывода информации.

Таким образом, в ходе работы процессора схема выборки команд выбирает последовательно команды из памяти, затем эти команды выполняются, причем в случае необходимости обработки данных подключается АЛУ. На входы АЛУ могут подаваться обрабатываемые данные из памяти или из внутренних регистров. Во внутренних регистрах хранятся также коды адресов обрабатываемых данных, расположенных в памяти. Результат обработки в АЛУ изменяет состояние регистра признаков и записывается во внутренний регистр или в память (как источник, так и приемник данных указывается в составе кода команды). При необходимости информация может переписываться из памяти (или из устройства ввода/вывода) во внутренний регистр или из внутреннего регистра в память (или в устройство ввода/вывода).

Внутренние регистры любого микропроцессора обязательно выполняют две служебные функции:

- определяют адрес в памяти, где находится выполняемая в данный момент команда (функция *счетчика команд* или *указателя команд*);
- определяют текущий адрес стека (функция *указателя стека*).

В разных процессорах для каждой из этих функций может отводиться один или два внутренних регистра. Эти два указателя отличаются от других не только своим специфическим, служебным, системным назначением, но и особым способом изменения содержимого. Их содержимое программы могут менять только в случае крайней необходимости, так как любая ошибка при этом грозит нарушением работы компьютера, зависанием и порчей содержимого памяти.

Содержимое ***указателя (счетчика) команд*** изменяется следующим образом. В начале работы системы (при включении питания) в него заносится раз и навсегда установленное значение. Это первый адрес программы начального запуска. Затем после выборки из памяти каждой следующей команды значение указателя команд автоматически увеличивается (инкрементируется) на единицу (или на два в зависимости от формата команд и типа процессора). То есть следующая команда будет выбираться из следующего по порядку адреса памяти. При выполнении команд перехода, нарушающих последовательный перебор адресов памяти, в указатель команд принудительно записывается новое значение — новый адрес в памяти, начиная с которого адреса команд опять же будут перебираться последовательно. Такая же смена содержимого указателя команд производится при вызове подпрограммы и возврате из нее или при начале обработки прерывания и после его окончания.

1.9 Лекция №10. Сетевая модель OSI (2 часа)

1.9.1 Вопросы лекции:

1. Сетевая модель OSI.
2. Структура стандартов IEEE 802.X.

1.9.2 Краткое содержание вопросов

1. Сетевая модель OSI.

Международная Организация по Стандартам (МОС, International Standards Organization – ISO) предложила в качестве стандарта открытых систем семиуровневую коммуникационную модель (рис.1.19), известную как OSI-модель (Open Systems Interconnection) – модель Взаимодействия Открытых Систем (ВОС).

Каждый уровень OSI-модели отвечает за отдельные специфические функции в коммуникациях и реализуется техническими и программными средствами вычислительной сети.

Физический уровень

Уровень 1 – физический (physical layer) – самый низкий уровень OSI-модели, определяющий процесс прохождения сигналов через среду передачи между сетевыми устройствами (узлами сети).

Реализует управление каналом связи:

- подключение и отключение канала связи;
- формирование передаваемых сигналов и т.п.

Описывает:

- механические, электрические и функциональные характеристики среды передачи;
- средства для установления, поддержания и разъединения физического соединения.

Обеспечивает при необходимости:

- кодирование данных;
- модуляцию сигнала, передаваемого по среде.

Данные физического уровня представляют собой поток битов (последовательность нулей или единиц), закодированные в виде электрических, оптических или радио сигналов.

Из-за наличия помех, воздействующих на электрическую линию связи, достоверность передачи, измеряемая как вероятность искажения одного бита, составляет 10-4 – 10 -6. Это означает, что в среднем на 10000 – 1000000 бит передаваемых данных один бит оказывается искажённым.

Канальный уровень

Канальный уровень или уровень передачи данных (data link layer) является вторым уровнем OSI-модели.

Реализует управление:

- доступом сетевых устройств к среде передачи, когда два или более устройств могут использовать одну и ту же среду передачи;
- надежной передачей данных в канале связи, позволяющей увеличить достоверность передачи данных на 2-4 порядка.

Описывает методы доступа сетевых устройств к среде передачи, основанные, например, на передаче маркера или на соперничестве.

Обеспечивает:

- функциональные и процедурные средства для установления, поддержания и разрыва соединения;
- управление потоком для предотвращения переполнения приемного устройства, если его скорость меньше, чем скорость передающего устройства;
- надежную передачу данных через физический канал с вероятностью искажения данных 10-8 – 10 -9 за счёт применения методов и средства контроля передаваемых данных и повторной передачи данных при обнаружении ошибки.

Таким образом, канальный уровень обеспечивает достаточно надежную передачу данных через ненадежный физический канал.

Блок данных, передаваемый на канальном уровне, называется кадром (frame).

На канальном уровне появляется свойство адресуемости

передаваемых данных в виде физических (машинных) адресов, называемых также MAC-адресами и являющихся обычно уникальными идентификаторами сетевых устройств.

Как будет показано в разделе 3, универсальные MAC-адреса в ЛВС Ethernet и Token Ring являются 6-байтными и записываются в шестнадцатеричном виде, причём байты адреса разделены дефисом, например: 00-19-45-A2-B4-DE .

К процедурам канального уровня относятся:

- добавление в кадры соответствующих адресов;
- контроль ошибок;
- повторная, при необходимости, передача кадров.

На канальном уровне работают ЛВС Ethernet, Token Ring и FDDI.

Сетевой уровень

Сетевой уровень (network layer), в отличие от двух предыдущих, отвечает за передачу данных в СПД и управляет маршрутизацией сообщений – передачей через несколько каналов связи по одной или нескольким сетям, что обычно требует включения в пакет сетевого адреса получателя.

Блок данных, передаваемый на сетевом уровне, называется пакетом (packet).

Сетевой адрес – это специфический идентификатор для каждой промежуточной сети между источником и приемником информации.

Сетевой уровень реализует:

- обработку ошибок,
- мультиплексирование пакетов;
- управление потоками данных.

Самые известные протоколы этого уровня:

- X.25 в сетях с коммутацией пакетов;
- IP в сетях TCP/IP;
- IPX/SPX в сетях NetWare.

Кроме того, к сетевому уровню относятся протоколы построения маршрутных таблиц для маршрутизаторов: OSPF, RIP, ES-IS, IS-IS.

Транспортный уровень

Транспортный уровень (transport layer) наиболее интересен из высших уровней для администраторов и разработчиков сетей, так как он управляет сквозной передачей сообщений между оконечными узлами сети ("end-end"), обеспечивая надежность и экономическую эффективность передачи данных независимо от пользователя. При этом оконечные узлы возможно взаимодействуют через несколько узлов или даже через несколько транзитных сетей.

На транспортном уровне реализуется:

1) преобразование длинных сообщений в пакеты при их передаче в сети и обратное преобразование;
2) контроль последовательности прохождения пакетов;
3) регулирование трафика в сети;
4) распознавание дублированных пакетов и их уничтожение.

Способ коммуникации "end-end" облегчается еще одним способом адресации – адресом процесса, который соотносится с определенной прикладной программой (прикладным процессом), выполняемой на компьютере. Компьютер обычно выполняет одновременно несколько программ, в связи с чем необходимо знать какой прикладной программе (процессу) предназначено поступившее сообщение. Для этого на транспортном уровне используется специальный адрес, называемый адресом порта. Сетевой уровень доставляет каждый пакет на конкретный адрес компьютера, а транспортный уровень передаёт полностью собранное сообщение конкретному прикладному процессу на этом компьютере.

Транспортный уровень может предоставлять различные типы сервисов, в частности, передачу данных без установления соединения или с предварительным установлением соединения. В последнем случае перед началом передачи данных с использованием

специальных управляющих пакетов устанавливается соединение с транспортным уровнем компьютера, которому предназначены передаваемые данные. После того как все данные переданы, подключение заканчивается. При передаче данных без установления соединения транспортный уровень используется для передачи одиночных пакетов, называемых дейтаграммами, не гарантируя их надежную доставку. Передача данных с установлением соединения применяется для надежной доставки данных.

Сеансовый уровень

Сеансовый уровень (session layer) обеспечивает обслуживание двух "связанных" на уровне представления данных объектов сети и управляет ведением диалога между ними путем синхронизации, заключающейся в установке служебных меток внутри длинных сообщений. Эти метки позволяют после обнаружения ошибки повторить передачу данных не с самого начала, а только с того места, где находится ближайшая предыдущая метка по отношению к месту возникновения ошибки.

Сеансовый уровень предоставляет услуги по организации и синхронизации обмена данными между процессами уровня представлений.

На сеансовом уровне реализуется:

1) установление соединения с адресатом и управление сеансом;
2) координация связи прикладных программ на двух рабочих станциях.

Уровень представления

Уровень представления (presentation layer) обеспечивает совокупность служебных операций, которые можно выбрать на прикладном уровне для интерпретации передаваемых и получаемых данных. Эти служебные операции включают в себя:

- управление информационным обменом;
- преобразование (перекодировка) данных во внутренний формат каждой конкретной ЭВМ и обратно;
- шифрование и дешифрование данных с целью защиты от несанкционированного доступа;
- сжатие данных, позволяющее уменьшить объём передаваемых данных, что особенно актуально при передаче мультимедийных данных, таких как аудио и видео.

Служебные операции этого уровня представляют собой основу всей семиуровневой модели и позволяют связывать воедино терминалы и средства вычислительной техники (компьютеры) самых разных типов и производителей.

Прикладной уровень

Прикладной уровень (application layer) обеспечивает непосредственную поддержку прикладных процессов и программ конечного пользователя, а также управление взаимодействием этих программ с различными объектами сети. Другими словами, прикладной уровень обеспечивает интерфейс между прикладным ПО и системой связи. Он предоставляет прикладной программе доступ к различным сетевым службам, включая передачу файлов и электронную почту.

2. Структура стандартов IEEE 802.X.

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

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

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

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

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

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

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

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

Стандарты IEEE 802 имеют достаточно четкую структуру, приведенную на рис. 16:

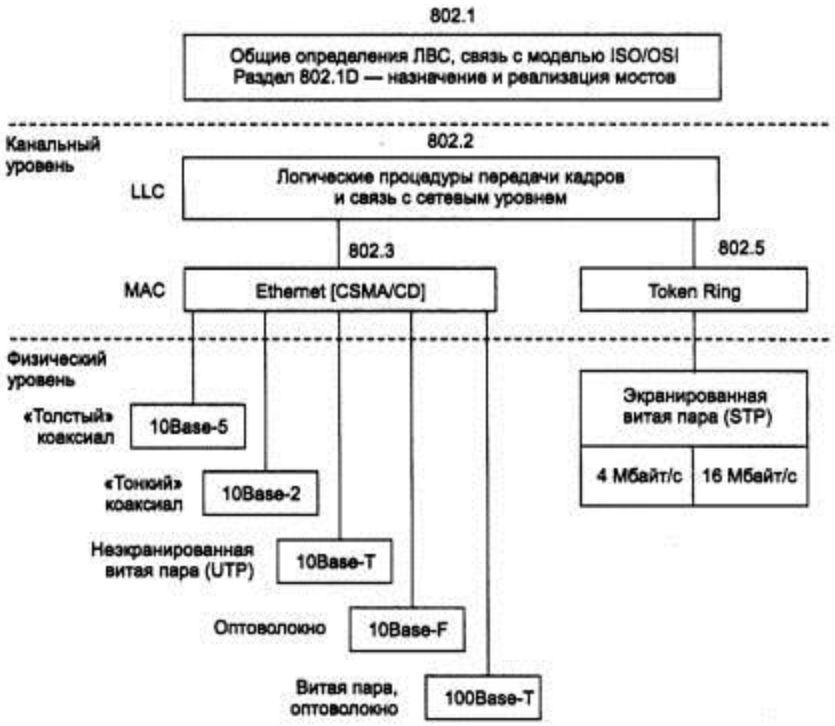


Рисунок 16. Структура стандартов IEEE 802.X

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

Над канальным уровнем всех технологий изображен общий для них протокол LLC, поддерживающий несколько режимов работы, но независимый от выбора конкретной технологии. Стандарт LLC курирует подкомитет 802.2. Даже технологии, стандартизованные не в рамках комитета 802, ориентируются на использование протокола LLC, определенного стандартом 802.2, например протокол FDDI, стандартизованный ANSI.

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

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

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

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

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

802.1 - Internetworking - объединение сетей;

802.2 - Logical Link Control, LLC - управление логической передачей данных;

802.3 - Ethernet с методом доступа CSMA/CD;

802.4 - Token Bus LAN - локальные сети с методом доступа Token Bus;

802.5 - Token Ring LAN - локальные сети с методом доступа Token Ring;

802.6 - Metropolitan Area Network, MAN - сетимегаполисов;

802.7 - Broadband Technical Advisory Group - техническая консультационная группа по широкополосной передаче;

802,8 - Fiber Optic Technical Advisory Group - техническая консультационная группа по волоконно-оптическим сетям;

802.9 - Integrated Voice and data Networks - интегрированные сети передачи голоса и данных;

802.10 - Network Security - сетевая безопасность;

802.11 - Wireless Networks - беспроводные сети;

802.12 - Demand Priority Access LAN, l00VG-AnyLAN - локальные сети с методом доступа по требованию с приоритетами.

1.10 Лекция №10. Запоминающие устройства ЭВМ (2 часа)

1.10.1 Вопросы лекции:

1. Запоминающие устройства ЭВМ

1.10.2 Краткое содержание вопросов:

1. Запоминающие устройства ЭВМ

Памятью ЭВМ называется совокупность устройств, служащих для запоминания, хранения и выдачи информации.

Отдельные устройства, входящие в эту совокупность, называются запоминающими устройствами (ЗУ) различных типов.

Термин "ЗАПОМИНАЮЩЕЕ УСТРОЙСТВО" обычно используется тогда, когда речь идет о принципе построения некоторого устройства памяти (например, полупроводниковое ЗУ, ЗУ на жестком магнитном диске и т.п.). А термин "ПАМЯТЬ" - когда хотят подчеркнуть выполняемую устройством памяти логическую функцию или место расположения в составе оборудования ЭВМ (например, оперативная память - ОП, внешняя память и т.п.).

Самое большое распространение запоминающие устройства приобрели в компьютерах (компьютерная память). Кроме того, они применяются в устройствах автоматики и телемеханики, в приборах для проведения экспериментов, в бытовых устройствах (телефонах, фотоаппаратах, холодильниках, стиральных машинах и т. д.), в пластиковых карточках, замках.

Запоминающие устройства играют важную роль в общей структуре ЭВМ. По некоторым оценкам производительность компьютера на разных классах задач на 40-50% определяется характеристиками ЗУ различных типов, входящих в его состав.

К основным параметрам, характеризующим запоминающие устройства, относятся емкость и быстродействие.

Емкость памяти - это максимальное количество данных, которое в ней может храниться.

Емкость запоминающего устройства измеряется количеством адресуемых элементов (ячеек) ЗУ и длиной ячейки в битах. В настоящее время практически все запоминающие устройства в качестве минимально адресуемого элемента используют 1 байт (1 байт = 8 двоичных разрядов (бит)) и емкость памяти указывается в укрупненных единицах (Кб, Мб, Гб и т.д.)

За одно обращение к запоминающему устройству производится считывание или запись некоторой единицы данных, называемой словом, различной для устройств разного типа.

Быстродействие памяти определяется продолжительностью операции обращения, то есть временем, затрачиваемым на поиск нужной информации в памяти и на ее считывание, или временем на поиск места в памяти, предназначаемого для хранения данной информации, и на ее запись.

Идеальное запоминающее устройство должно обладать бесконечно большой емкостью и иметь бесконечно малое время обращения. На практике эти параметры находятся в противоречии друг другу: в рамках одного типа ЗУ улучшение одного из них ведет к ухудшению значения другого. К тому же следует иметь в виду и экономическую целесообразность построения запоминающего устройства с теми или иными характеристиками при данном уровне развития технологии. Поэтому в настоящее время запоминающие устройства компьютера, как это и предполагал Нейман, строятся по иерархическому принципу.

Иерархическая структура памяти позволяет экономически эффективно сочетать хранение больших объемов информации с быстрым доступом к информации в процессе ее обработки.

В соответствии с принципом иерархии памяти выделяют внутреннюю и внешнюю память компьютера. Первая используется для временного хранения данных и программ при выполнении последних, а вторая - для долговременного хранения данных и программ.

Внутренняя память компьютера (ее еще называют: основная память) предназначена для оперативной обработки данных. Она является более быстрой, чем внешняя память, что соответствует принципу иерархии памяти, выдвинутому в проекте Принстонской машины. Следуя этому принципу, можно выделить уровни иерархии и во внутренней памяти.

На нижнем уровне иерархии находится регистровая память - набор регистров, входящих непосредственно в состав микропроцессора (центрального процессора - CPU). Регистры CPU программно доступны и хранят информацию, наиболее часто используемую при выполнении программы: промежуточные результаты, счетчики циклов и т.д. Регистровая память имеет относительно небольшой объем (до нескольких десятков машинных слов). РП работает на частоте процессора, поэтому время доступа к ней минимально и измеряется в нс.

Оперативная память (ОЗУ) - устройство, которое служит для хранения информации (программ, исходных данных, промежуточных и конечных результатов обработки), непосредственно используемой в ходе выполнения программы в процессоре. В настоящее время объем ОП персональных компьютеров составляет несколько сотен мегабайт. Время обращения к оперативной памяти составляет наносекунды.

Для заполнения пробела между РП и ОП по объему и времени обращения в настоящее время используется кэш-память, высокоскоростная память (дорогая), является буфером между ОП и микропроцессором, статическая оперативная память со специальным механизмом записи и считывания информации и предназначена для хранения информации, наиболее часто используемой при работе программы. Как правило, часть кэш-памяти располагается непосредственно на кристалле микропроцессора (внутренний кэш), а часть - вне микропроцессора (внешняя кэш-память). Кэш-память программно недоступна. Для обращения к ней используются аппаратные средства процессора и компьютера.

ПЗУ в англоязычной литературе называется Read Only Memory (ROM), что дословно переводится как "память только для чтения". Название ПЗУ говорит само за себя. Информация в ПЗУ записывается на заводе-изготовителе микросхем памяти, и в дальнейшем изменить ее значение нельзя. В ПЗУ хранится критически важная для компьютера информация, которая не зависит от выбора операционной системы. ПЗУ - энергонезависимая память.

В ПЗУ находятся:

Программы запуска и остановки ЭВМ

Программы тестирования устройств, проверяющие при каждом включении компьютера правильность работы его блоков

Программа управления работой самого процессора

Программы управления дисплеем, клавиатурой, принтером, внешней памятью

Информация о том, где на диске находится операционная система.

Внешняя (долговременная) память - это место длительного хранения данных (программ, результатов расчётов, текстов и т.д.), не используемых в данный момент в оперативной памяти компьютера. Внешняя память, в отличие от оперативной, является энергонезависимой. Носители внешней памяти, кроме того, обеспечивают транспортиров-ку данных (перенос с компьютера на компьютер) в тех случаях, когда компьютеры не объединены в сети (локальные или глобальные). Внешняя память организуется, как правило, на магнитных и оптических дисках, магнитных лентах.

Для работы с внешней памятью необходимо наличие накопителя (устройства, обеспечивающего запись и (или) считывание информации) и устройства хранения информации - носителя.

Основные виды накопителей:

накопители на жестких магнитных дисках (НЖМД);

накопители на оптических дисках (НОД);

накопители на магнитной ленте (НМЛ);

накопители на гибких магнитных дисках (НГМД); и др.

Им соответствуют основные виды носителей:

жёсткие магнитные диски (Hard Disk);

диски CD-ROM, CD-R, CD-RW, DVD.

кассеты для стримеров и других МЛ;

гибкие магнитные диски (Floppy Disk) (диаметром 3,5дюйма и ёмкостью 1,44 Мб в настоящее время устарели и практически не используются) и др.

Накопители на жестких магнитных дисках (НЖМД).

Накопитель на жёстких магнитных дисках (винчестер) - это наиболее массовое запоминающее устройство большой ёмкости, в котором носителями информации являются круглые алюминиевые пластины – платтеры, обе поверхности которых покрыты слоем магнитного материала. На этой магнитной поверхности диска хранятся данные. Информация записывается и снимается с помощью магнитных головок. Используется для постоянного хранения информации: программ и данных.

Так как НЖМД имеют большие емкость и скорость обмена, они составляют основу внешней памяти компьютеров, которая содержит базовое системное и прикладное ПО, а также данные для оперативной обработки. Является основным накопителем данных практически во всех современных компьютерах.

Накопители на жестких дисках объединяют в одном корпусе носитель (носители) и устройство чтения/записи, а также, нередко, и интерфейсную часть, называемую контроллером жесткого диска. Типичной конструкцией жесткого диска является исполнение в виде одного устройства - камеры, внутри которой находится один или более дисковых носителей, помещённых на одну ось, и блок головок чтения/записи с их общим приводящим механизмом.

Принцип работы магнитных запоминающих устройств основан на способах хранения информации с использованием магнитных свойств материалов. Общая технология магнитных запоминающих устройств состоит в намагничивании переменным магнитным полем участков носителя и считывания информации, закодированной как области переменной намагниченности. Дисковые носители, как правило, намагничиваются вдоль концентрических полей – дорожек, расположенных по всей плоскости вращающегося носителя. Запись производится в цифровом коде. Плоский дисковый носитель вращается в процессе чтения/записи, чем и обеспечивается обслуживание всей концентрической дорожки, чтение и запись осуществляется при помощи магнитных головок чтения/записи, которые позиционируют по радиусу носителя с одной дорожки на другую.

Для операционной системы данные на дисках организованы в дорожки и секторы. Дорожки (40 или 80) представляют собой узкие концентрические кольца на диске. Каждая дорожка разделена на части, называемые секторами. При чтении или записи устройство всегда считывает или записывает целое число секторов независимо от объёма запрашиваемой информации. Размер сектора равен 512 байт. Цилиндр — это общее количество дорожек, с

которых можно считать информацию, не перемещая головок. Кластер - наименьшая область диска, которую операционная система использует при записи файла. Обычно кластер - один или несколько секторов. В ПК обычно устанавливается один или два накопителя на гибких дисках и один накопитель на жестких дисках.

Винчестеры имеют следующие основные характеристики:

Емкость жёстких дисков – в настоящее время превышает 300 Гб.

Число поверхностей — определяет количество физических дисков, нанизанных на ось.

Число цилиндров — определяет, сколько дорожек будет располагаться на одной поверхности.

Число секторов — на одной дорожке и общее число секторов на всех дорожках всех поверхностей накопителя.

время доступа к данным;

Наряду с НЖМД, но значительно реже, используются и накопители на магнитных барабанах (НМБ).

Накопителям, для их различения, присваиваются имена (идентификаторы). За накопителями на гибких дисках зарезервированы имена A:, B:, за накопителями на жестких дисках - имена C: и D:.

НАКОПИТЕЛИ НА ОПТИЧЕСКИХ ДИСКАХ (НОД)

В последнее время все более широкое распространение получают накопители на оптических дисках (НОД). Благодаря маленьким размерам (компакт диски диаметром 3,5 и 5.25 дюйма), большой емкости и надежности они становятся все более популярными, все интенсивнее используются на ПК.

Принцип действия оптических дисков основан на использовании лазера для записи и чтения информации. В процессе записи модулированный по интенсивности луч лазера оставляет на диске след, который потом можно прочитать, направив на него луч лазера и проанализировав отраженный луч.

По функциональному признаку НОД делятся:

1. НОД только для чтения;
2. НОД с однократной записью и многократным чтением;
3. НОД с возможностью перезаписи.

Первая категория НОД использует технологию CD-ROM для записи звука. Компакт-диски подобно пластинкам изготавливаются на заводе изготовителе и распространяются с нанесенной на них информацией, предназначенной только для чтения. CD-ROM имеет очень плотную запись информации и емкость диска достигает 2 Гбайт.

Вторая категория дисков позволяет самостоятельно записать один раз информацию, однако затем записанную информацию невозможно ни стереть, ни перезаписать. Сущность процесса записи состоит в том, что лазерный луч разогревает поверхность диска перед магнитной записывающей головкой. Поверхность диска меняет отражательную способность, что обнаруживает читающий луч лазера. Емкость таких дисков доходит до 10 Гбайт.

Для третьей категории оптических дисков принцип записи и чтения аналогичен НОД второй категории, но в отличие от них эти диски допускают многократную запись информации.

CD-ROM - НОД представляют интерес для архивирования информации, и в качестве удобного средства ее транспортировки; хранения словарей, справочников, распространения лицензионного ПО, а в настоящее время широко распространена продажа нелицензионного ПО именно на томах CD-ROM. это компакт-диск, оптический носитель информации, предназначенный только для чтения. Доступ к данным на CD-ROM осуществляется быстрее, чем к данным на дискетах, но медленнее, чем на жёстких дисках. По причине большой емкости и способа использования компакт-диск изготовлен из полимера и покрыт металлической плёнкой. Информация считывается именно с этой металлической плёнки, которая покрывается полимером, защищающим данные от повреждения. CD-ROM является односторонним носителем информации.

Считывание информации с диска происходит с помощью лазера. Фотодатчик воспринимает рассеянный луч, и эта информация в виде электрических сигналов поступает на микропроцессор, который преобразует эти сигналы в двоичные данные.

Накопители CD-R (CD-Recordable) позволяют записывать собственные компакт-диски.

Накопители CD-RW - являются наиболее популярными, позволяют записывать и перезаписывать диски CD-RW, записывать диски CD-R, читать диски CD-ROM, т.е. являются в определённом смысле универсальными компакт-дисками многократной перезаписи, ресурс перезаписи одного диска составляет примерно несколько тысяч циклов.

Диски однократной и многократной записи внешне они ничем не отличаются от обычных компакт-дисков, но устройство у них разное.

DVD-ROM и DVD-RW. Снаружи, диски DVD выглядят также как и CD-диски. Однако возможностей у DVD гораздо больше. Диски DVD могут хранить в 26 раз больше данных, по сравнению с обычным CD-ROM. Имея физические размеры и внешний вид, как у обычного компакт-диска или CD-ROM, диски DVD стали огромным скачком в области емкости для хранения информации. По сравнению со своим предком, вмещающим 700 MB данных, стандартный однослойный, односторонний диск DVD может хранить 4.7GB данных. Но это не предел - DVD могут изготавливаться по двухслойному стандарту, который позволяет увеличить емкость хранимых на одной стороне данных до 8.5GB. Кроме этого, диски DVD могут быть двухсторонними, что увеличивает емкость одного диска до 17GB. Преимущества нового стандарта максимально проявляются как раз при просмотре видеофильмов: по качеству изображения DVD-диск на голову превосходит все предыдущие видеоформаты, значительное повышение качества звука обеспечивается применением новых цифровых систем многоканального звука. Наряду с повышением качества добавляется ряд абсолютно новых, не свойственных другим форматам, интерактивных возможностей: выбор языка озвучивания, включение и выбор языка субтитров, выбор ракурса просмотра сцены, выбор варианта развития событий, быстрое переключение между сценами.

Blu-Ray BD (англ. blue ray — синий луч и disc — диск) — формат оптического носителя, используемый для записи и хранения цифровых данных, включая видео высокой чёткости с повышенной плотностью.

НАКОПИТЕЛИ НА МАГНИТНЫХ ЛЕНТАХ (НМЛ)

Накопители на магнитных лентах были первыми внешними запоминающим устройствами и широко использовались и используются на больших ЭВМ. На ПЭВМ накопители на магнитных лентах не получили широкого распространения. Они используются только, как устройства для резервирования (архивации) информации.

Лентопротяжные механизмы носят название стриммеры. Кассеты с магнитной лентой весьма разнообразны. Емкость кассет значительна и достигает десятки Гбайт. Ленты относятся к носителям информации с последовательным доступом. Последовательный доступ означает, что вычислительная машина может обратиться к необходимой записи только перематывая ленту к тому месту где она располагается, т. е. просматривая записи последовательно друг за другом.

Достоинство магнитных лент является небольшая стоимость и очень высокая надежность хранения информации. Серьезным конкурентом магнитным лентам несомненно выступают оптические диски с многократной перезаписью информации.

Обычно, НМЛ используются в мини-, супер- и общего назначения ЭВМ для архивного хранения данных и программ, как устройства резервного копирования данных, обращение к которым происходит редко, а может быть и никогда, ибо последовательный метод доступа к ним делает нецелесообразным использование их в качестве внешней памяти оперативного обмена. Наряду с традиционным оформлением магнитных лент в виде бобин, используются картриджи, стримеры (особенно для ПК, запись производится на мини-кассеты). Стримеры позволяют записать на небольшую кассету с магнитной лентой до 13 Гб информации. Встроенные в стример средства аппаратного сжатия позволяют автоматически уплотнять информацию перед её записью и восстанавливать после считывания, что увеличивает объём

сохраняемой информации. Недостатком стримеров является их сравнительно низкая скорость записи, поиска и считывания информации.

Накопители на гибких магнитных дисках (НГМД).

Гибкий диск, дискета - устройство для хранения небольших объёмов информации, представляющее собой гибкий пластиковый диск с магнитным покрытием в защитной оболочке (квадратный предохранительный конверт). Дискета устанавливается в накопи-тель на гибких магнитных дисках, автоматически в нем фиксируется. В накопителе вращается сама дискета, магнитные головки остаются неподвижными. Дискета вращается только при обращении к ней. Накопитель связан с процессором через контроллер гибких дисков.

Информация записывается по концентрическим дорожкам (трекам), которые делятся на секторы. Количество дорожек и секторов зависит от типа и формата дискеты. Сектор хранит минимальную порцию информации, которая может быть записана на диск или считана. Ёмкость сектора постоянна и составляет 512 байт. Последние используемые дискеты ёмкостью 1,44 Мб и диаметром 3,5дюйма в настоящее время устарели и практически не применяются.

Флэш-память - особый вид энергонезависимой, перезаписывае-мой полупроводниковой памяти. Энергонезависимая - не требующая дополнительной энергии для хранения данных (энергия требуется только для записи), перезаписываемая - допускающая изменение (перезапись) хранимых в ней данных и полупроводниковая (твердотелая) то есть не содержащая механически движущихся частей (как обычные жёсткие диски или CD), построенная на основе интегральных микросхем.

Флэш-память исторически происходит от ROM памяти, и функционирует подобно RAM. При отключении питания данные из флэш-памяти не пропадают. Flash-память, работает существенно медленнее ОП и имеет ограничение по количеству циклов перезаписи (от 10.000 до 1.000.000 для разных типов).

Информация, записанная на флэш-память, может храниться длительное время (от 20 до 100 лет), и способна выдерживать значительные механические нагрузки (в 5-10 раз превышающие предельно допустимые для обычных жёстких дисков). Основные преимущества Flash-памяти перед жёсткими дисками и носителями CD-ROM: Flash-память потребляет значительно (примерно в 10-20 и более раз) меньше энергии во время работы. В устройствах CD-ROM, жёстких дисках, кассетах и других механических носителях информации, большая часть энергии уходит на приведение в движение механики этих устройств. Кроме того, флэш-память компактнее большинства других механических носителей. Благодаря низкому энергопотреблению, компактности, долговечности и относительно высокому быстродействию, флэш-память идеально подходит для использования в качестве накопителя в таких портативных устройствах, как: цифровые фото- и видео камеры, сотовые телефоны, портативные компьютеры, MP3-плееры, цифровые диктофоны, и т.п.

Физические основы функционирования

К настоящему времени создано множество устройств, предназначенных для хранения данных, основанных на использовании самых разных физических эффектов. Универсального решения не существует, каждое содержит те или иные недостатки. Поэтому компьютерные системы обычно оснащаются несколькими видами систем хранения, основные свойства которых обуславливают их использование и назначение

В основе работы запоминающего устройства может лежать любой физический эффект, обеспечивающий приведение системы к двум или более устойчивым состояниям. В современной компьютерной технике часто используются физические свойства полупроводников (флэш-память), когда прохождение тока через полупроводник или его отсутствие трактуются как наличие логических сигналов 0 или 1. Устойчивые состояния, определяемые направлением намагниченности (МЛ, МД), позволяют использовать для хранения данных разнообразные магнитные материалы. Отражение или рассеяние света от поверхности (CD, DVD или Blu-ray диска) также позволяет хранить информацию.

1.11 Лекция №11. Протоколы и алгоритмы маршрутизации (2 часа)

1.11.1 Вопросы лекции:

1. Классификация протоколов маршрутизации.
2. Алгоритмы маршрутизации.
3. Внешние и внутренние шлюзовые протоколы.

1.11.2 Краткое содержание вопросов:

1.Классификация протоколов маршрутизации

Протоколы маршрутизации предназначены для автоматического построения таблиц маршрутизации, на основе которых происходит продвижение пакетов сетевого уровня. Протоколы маршрутизации, в отличие от сетевых протоколов, таких как IP и IPX, не являются обязательными, так как таблица маршрутизации может быть построена администратором сети вручную. При этом в крупных сетях со сложной топологией и большим количеством альтернативных маршрутов протоколы маршрутизации выполняют очень важную и полезную работу, автоматизируя построение таблиц маршрутизации, динамически адаптируя текущий набор рабочих маршрутов к состоянию сети и повышая тем самым ее производительность и надежность.

В настоящее время существует ряд протоколов обеспечивающих маршрутизацию в Ad-Hoc сетях. По принципу поиска маршрута выделяют три класса протоколов: проактивные, реактивные и комбинированные.

Проактивные протоколы маршрутизации

Эти типы протоколов обеспечивают предварительное построение таблице маршрутизации, в которую включаются все известные маршруты. Подобный подход используют все протоколы маршрутизации в проводных локальных сетях. В этом случае, пересылка пакетов начинается без задержек, но присутствуют накладные расходы на поиск маршрутов и построение таблице маршрутизации, потому что необходимо получить всю необходимую информацию о топологии сети до начала передачи пакетов. В дальнейшем данные протоколы предполагают дополнительные расходы на поддержание маршрутной информации в актуальном виде. Примерамиявляются DSDV (Destination Sequenced Distance Vector), OLSR (Optimized Link State Routing).

Реактивные протоколы маршрутизации

Эти типы протоколов называются протоколами по требованию. Узлы не хранят необходимую маршрутную информацию все время. Узел инициирует построение маршрута по требованию, в момент поступления запроса на передачу данных. Этот механизм построения маршрута основан на алгоритме наводнения, узел просто передает первый пакет всем своим соседям, а промежуточные узлы перенаправляют его далее, к своим соседям. Это повторяющиеся действия позволяют доставить пакет до пункта назначения. Как правило, при прохождении пакета запоминается маршрут прохождения (например, в виде списка задействованных узлов) и впоследствии при передаче последующих пакетов эта информация используются для выбора направления. Реактивные методы имеют меньшие накладные расходы на маршрутизацию, но большую задержку при инициации передачи, потому что маршрут между узлами будет найден только тогда, когда один из узлов получит запрос на передачу. Примеромреактивныхпротоколовявляются DSR(Dynamic Source Routing), AODV(Ad hoc On-Demand Distance Vector), TOPA(Temporally ordered routing algorithm).

Гибридные протоколы маршрутизации

Гибридные протоколы являются комбинации реактивного и проактивного подходов. Они используют преимущества этих двух протоколов и, как следствие, достаточно эффективно работают в отдельных случаях. Примером гибридного протокола может являться ZRP (Zone Routing Protocol) и HWMP (Hybrid Wireless Mesh Protocol).

2. Алгоритмы маршрутизации.

Алгоритмы маршрутизации применяются для определения оптимального пути пакетов от источника к получателю и являются основой любого протокола маршрутизации.

Типы алгоритмов

Алгоритмы маршрутизации могут быть классифицированы по типам:

•Статические или динамические. Статические алгоритмы представляют свод правил работы со статическими таблицами маршрутизации, которые настраиваются администраторами сети. Хорошо работают в случае предсказуемого трафика в сетях стабильной конфигурации. Динамические алгоритмы маршрутизации подстраиваются к изменяющимся обстоятельствам сети в масштабе реального времени. Они выполняют это путем анализа поступающих сообщений об обновлении маршрутизации. Если в сообщении указывается, что имело место изменение сети, программы маршрутизации пересчитывают маршруты и рассылают новые сообщения о корректировке маршрутизации. Такие сообщения пронизывают сеть, стимулируя маршрутизаторы заново прогонять свои алгоритмы и соответствующим образом изменять таблицы маршрутизации. Динамические алгоритмы маршрутизации могут дополнять, где это уместно, статические маршруты.

•Одномаршрутные или многомаршрутные алгоритмы. Некоторые сложные протоколы маршрутизации обеспечивают множество маршрутов к одному и тому же пункту назначения. Такие многомаршрутные алгоритмы делают возможной мультиплексную передачу трафика по многочисленным линиям, одномаршрутные алгоритмы не могут делать этого. Многомаршрутные алгоритмы могут обеспечить значительно большую пропускную способность и надежность.

•Одноуровневые или иерархические алгоритмы. Отличаются по принципу взаимодействия друг с другом. В одноуровневой системе маршрутизации все рутеры равны по отношению друг к другу. В иерархической системе маршрутизации пакеты данных перемещаются от роутеров нижнего уровня к базовым, которые осуществляют основную маршрутизацию. Как только пакеты достигают общей области пункта назначения, ониперемежаются вниз по иерархии до хоста назначения.

•Алгоритмы с маршрутизацией от источника. В системах маршрутизации от источника роутеры действуют просто как устройства хранения и пересылки пакета, безвсякий раздумий отсылая его к следующей остановке, они предполагают, что отправитель рассчитывает и определяет весь маршрут сам. Другие алгоритмы предполагают, что хост отправителя ничего не знает о маршрутах. При использовании такого рода алгоритмов роутеры определяют маршрут через сеть, базируясь на своих собственных расчетах.

•Внутридоменные или междоменные алгоритмы. Некоторые алгоритмы маршрутизации действуют только в пределах доменов; другие - как в пределах доменов, так и между ними.

•Алгоритмы состояния канала и дистанционно-векторные. Алгоритмы состояния канала направляют потоки маршрутной информации во все узлы сети. Каждый роутер отсылает только ту часть известной ему информации, которая описывает состояние его собственных каналов, но всем узлам маршрутизации. Дистанционно-векторные требуют от каждого роутера пересылки всей или части его таблицы но только соседям.

Качество алгоритма определяется следующими показателями:

- Оптимальность,
- Простота и низкие непроизводительные затраты,
- Живучесть и стабильность,
- Быстрая сходимость,
- Гибкость.

Оптимальность

Оптимальность характеризует способность алгоритма маршрутизации выбирать "наилучший" маршрут. Наилучший маршрут зависит от показателей и от "веса" этих показателей, используемых при проведении расчета.

Простота и низкие непроизводительные затраты

Алгоритмы маршрутизации разрабатываются как можно более простыми, чтобы эффективно обеспечивать свои функциональные возможности, с мимимальными затратами

программного обеспечения. Особенно это важно, когда маршрутизация, должна выполняться в компьютере с ограниченными физическими ресурсами.

Живучесть и стабильность

Алгоритмы маршрутизации должны обладать живучестью. Другими словами, они должны четко функционировать в случае неординарных или непредвиденных обстоятельств, таких как отказы аппаратуры, условия высокой нагрузки и некорректные реализации. Т.к. роутеры расположены в узловых точках сети, их отказ может вызвать значительные проблемы.

Часто наилучшими алгоритмами маршрутизации оказываются те, которые выдержали испытание временем и доказали свою надежность в различных условиях работы сети.

Быстрая сходимость

Алгоритмы маршрутизации должны быстро сходиться. Сходимость - это процесс соглашения между всеми роутерами по оптимальным маршрутам. Когда какое-нибудь событие в сети приводит к тому, что маршруты или отвергаются, или наоборот, становятся доступными, роутеры рассылают сообщения об обновлении маршрутизации. Такие сообщения пронизывают сети, стимулируя пересчет оптимальных маршрутов и, в конечном итоге, вынуждая все роутеры придти к соглашению по этим маршрутам. Алгоритмы маршрутизации, которые сходятся медленно, могут привести к образованию петель маршрутизации или выходам из строя сети.

Гибкость

Алгоритмы маршрутизации должны быть также гибкими, т.е. быстро и точно адаптироваться к разнообразным обстоятельствам в сети, таким как изменения полосы пропускания сети, размеров очереди к роутеру, величины задержки сети и других переменных. Например, предположим, что сегмент сети отвергнут. Многие алгоритмы маршрутизации, после того как они узнают об этой проблеме, быстро выбирают следующий наилучший путь для всех маршрутов, которые обычно используют этот сегмент.

Маршрутные таблицы содержат информацию, которую используют программы для выбора наилучшего маршрута. Ниже перечислены показатели, которые используются в алгоритмах маршрутизации:

- Длина маршрута,
- Надежность,
- Задержка,
- Ширина полосы пропускания,
- Нагрузка,
- Стоимость связи.

Сложные алгоритмы маршрутизации при выборе маршрута могут базироваться на множестве показателей, комбинируя их таким образом, что в результате получается один отдельный (гибридный) показатель.

Длина маршрута

Длина маршрута является наиболее общим показателем маршрутизации. Некоторые протоколы маршрутизации позволяют администраторам сети назначать произвольные цены на каждый канал сети. В этом случае длиной тракта является сумма расходов, связанных с каждым каналом. Другие протоколы маршрутизации определяют "количество пересылок", т.е. показатель, характеризующий число проходов, которые пакет должен совершить на пути от источника до пункта назначения через изделия об'единения сетей (такие как роутеры).

Надежность

Надежность каждого канала сети в контексте алгоритмов маршрутизации обычно описывается в терминах отношения бит/ошибка. Некоторые каналы сети могут отказывать

чаще, чем другие. Отказы одних каналов сети могут быть устранены легче или быстрее, чем отказы других каналов. При назначении оценок надежности могут быть приняты в расчет любые факторы надежности. Оценки надежности обычно назначаются каналам сети администраторами сети. Как правило, это произвольные цифровые величины.

Задержка

Под задержкой маршрутизации обычно понимают отрезок времени, необходимый для передвижения пакета от источника до пункта назначения через об'единенную сеть. Задержка зависит от многих факторов, включая полосу пропускания промежуточных каналов сети, очереди в порт каждого роутера на пути передвижения пакета, перегруженность сети на всех промежуточных каналах сети и физическое расстояние, на которое необходимо переместить пакет. Т.к. здесь имеет место конгломерация нескольких важных переменных, задержка является наиболее общим и полезным показателем.

Полоса пропускания

Полоса пропускания относится к имеющейся мощности трафика какого-либо канала. При прочих равных показателях, канал Ethernet 10 Mbps предпочтителен любой арендованной линии с полосой пропускания 64 Кбайт/сек. Хотя полоса пропускания является оценкой максимально достижимой пропускной способности канала, маршруты, проходящие через каналы с большей полосой пропускания, не обязательно будут лучше маршрутов, проходящих через менее быстродействующие каналы.

3. Внешние и внутренние шлюзовые протоколы.

Такое решение было найдено для самой крупной на сегодня составной сети — Интернета. Это решение базируется на понятии автономной системы.

Обычно автономной системой управляет один поставщик услуг Интернета, самостоятельно выбирая, какие протоколы маршрутизации должны использоваться в некоторой автономной системе и каким образом между ними должно выполняться перераспределение маршрутной информации. Крупные поставщики услуг и корпорации могут представить свою составную сеть как набор нескольких автономных систем. Регистрация автономных систем происходит централизованно, как и регистрация IP-адресов и DNS-имен. Номер автономной системы состоит из 16 разрядов и никак не связан с префиксами IP-адресов входящих в нее сетей.

В соответствии с этой концепцией Интернет выглядит как набор взаимосвязанных автономных систем, каждая из которых состоит из взаимосвязанных сетей (рис. 20), соединенными внешними шлюзами.

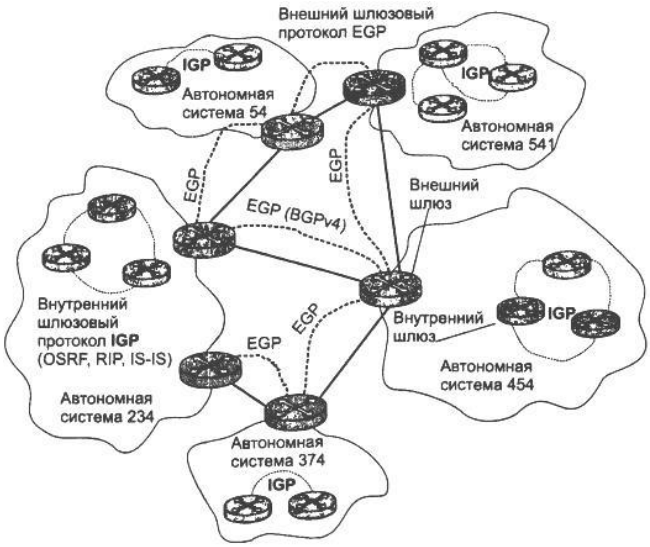


Рисунок 20 Автономные системы Интернета

Основная цель деления Интернета на автономные системы — обеспечение многоуровневого подхода к маршрутизации. До введения автономных систем предполагался двухуровневый подход, то есть сначала маршрут определялся как последовательность сетей, а затем вел непосредственно к заданному узлу в конечной сети (именно этот подход мы использовали до сих пор).

С появлением автономных систем появляется третий, верхний, уровень маршрутизации — теперь сначала маршрут определяется как последовательность автономных систем, затем — как последовательность сетей и только потом ведет к конечному узлу.

Выбор маршрута между автономными системами осуществляют внешние шлюзы, использующие особый тип протокола маршрутизации, так называемый внешний шлюзовой протокол (Exterior Gateway Protocol, EGP). В настоящее время для работы в такой роли сообщество Интернета утвердило стандартный пограничный шлюзовой протокол версии 4 (Border Gateway Protocol, BGPv4). В качестве адреса следующего маршрутизатора в протоколе BGPv4 указывается адрес точки входа в соседнюю автономную систему.

За маршрут внутри автономной системы отвечают внутренние шлюзовые протоколы (Interior Gateway Protocol, IGP). К числу IGP относятся знакомые нам протоколы RIP, OSPF и IS-IS. В случае транзитной автономной системы эти протоколы указывают точную последовательность маршрутизаторов от точки входа в автономную систему до точки выхода из нее.

Внутри каждой автономной системы может применяться любой из существующих протоколов маршрутизации, в то время как между автономными системами всегда применяется один и тот же протокол, являющийся своеобразным языком «эсперанто», на котором автономные системы общаются между собой.

Концепция автономных систем скрывает от администраторов магистрали Интернета проблемы маршрутизации пакетов на более низком уровне — уровне сетей. Для администратора магистрали неважно, какие протоколы маршрутизации применяются внутри автономных систем, для него существует единственный протокол маршрутизации — BGPv4.

1.12 Лекция №13. Устройства ввода вывода информации в ЭВМ.

1.12.1 Вопросы лекции:

1. Изучить технологию ATMI;
2. Ознакомиться с особенностями ATM.

1.12.2 Краткое содержание вопросов:

Устройства ввода и вывода - устройства взаимодействия компьютера с внешним миром: с пользователями или другими компьютерами. Устройства ввода позволяют вводить информацию в компьютер для дальнейшего хранения и обработки, а устройства вывода - получать информацию из компьютера.

Устройства ввода и вывода относятся к периферийным (дополнительным) устройствам.

Периферийные устройства - это все устройства компьютера, за исключением процессора и внутренней памяти.

Классификация периферийных устройств по месту расположения (относительного системного блока настольного компьютера или корпуса ноутбука):

внутренние - находятся внутри системного блока\корпуса ноутбука: жесткий диск (винчестер), встроенный дисковод (привод дисков);

внешние - подключаются к компьютеру через порты ввода-вывода: мышь, принтер и т.д.

По другому определению, периферийными устройствами называют устройства, не входящие в системный блок компьютера.

Устройства ввода и вывода разделяются на:

устройства ввода,
устройства вывода,
устройства ввода-вывода.
Устройства ввода данных

Классификация по типу вводимой информации:
устройства ввода текста: клавиатура;
устройства ввода графической информации:
сканер,
цифровые фото- и видеокамера,
веб камера - цифровая фото- или видеокамера маленького размера, которая делает фото или записывает видео в реальном времени для дальнейшей их передачи по сети Интернет;
графический планшет (дигитайзер) - для ввода чертежей, графиков и планов с помощью специального карандаша, которым водят по экрану планшета;
устройства ввода звука: микрофон;
Устройства-манипуляторы (преобразуют движение руки в управляющую информацию для компьютера):
несенсорные:
мышь,
трекбол - устройство в виде шарика, управляется вращением рукой;
трекпойнт (Pointing stick) - джойстик очень маленького размера (5 мм) с шершавой вершиной, который расположен между клавишами клавиатуры, управляется нажатием пальца;
игровые манипуляторы: джойстик, педаль, руль, танцевальная платформа, игровой пульт (геймпад, джойпад);
сенсорные:
тачпад (сенсорный коврик) - прямоугольная площадка с двумя кнопками, управляется движением пальца и нажатием на кнопки, используется в ноутбуках,
сенсорный экран - экран, который реагирует на прикосновение пальца или стилуса (палочка со специальным наконечником), используется в планшетных персональных компьютерах;
графический планшет (дигитайзер) - для ввода чертежей, схем и планов с помощью специального карандаша, которым водят по экрану планшета,
световое перо - устройство в виде ручки, ввод данных приконовением или проведением линий по экрану ЭЛТ-монитора (монитора на основе электронно-лучевой трубки). Сейчас световое перо не используется.
Устройства вывода данных
Классификация по типу выводимой информации:
устройства вывода графической и текстовой информации:
монитор - для вывода на дисплей (экран монитора),
проектор - для вывода на большой экран,
устройства для вывода на печать:
принтер - для вывода информации на бумагу, а также на поверхность дисков;
широкоформатный принтер ("широкий" принтер) - для вывода на листах форматов: A0, A1, A2 и A3,
плоттер (графопостроитель) - для вывода векторных изображений (различных чертежей и схем) на бумаге, картоне, кальке;
каттер (режущий плоттер) - вырезает изображения из пленки, картона по заданному контуру;
устройства вывода (воспроизведения) звука :
наушники,
колонки и акустические системы (динамик, усилитель),
встроенный динамик (PC speaker; Beeper) - для подачи звукового сигнала в случае возникновения ошибки.

Устройства ввода-вывода:

жесткий диск (винчестер) (входящий в него дисковод) - для ввода-вывода информации на жесткие пластины жесткого диска;

флэшка (флешка или USB-флеш-накопитель) - для ввода-вывода информации на микросхему памяти флэшки

дисководод оптических дисков - для ввода-вывода информации на оптические диски,

дисководод гибких дисков - для ввода-вывода информации на дискеты,

стример - для ввода-вывода информации на картриджи (ленточные носители);

кардридер - для ввода-вывода информации на карту памяти;

многофункциональное устройство (МФУ) - копировальный аппарат с дополнительными функциями принтера (вывод данных) и сканера (ввод данных)

модем (телефонный) - для связи компьютеров через телефонную сеть;

сетевая плата (сетевая карта или сетвой адаптер) - для подключения персонального компьютера к сети и организации взаимодействия с другими устройствами сети (обмен информацией по сети).

1.13 Лекция №14. Виды архитектур ЛВС.

1.13.1 Вопросы лекции:

1. Изучить технологию ATMI;
2. Ознакомиться с особенностями ATM.

1.13.2 Краткое содержание вопросов:

Вычислительная сеть (ВС) – это сложный комплекс взаимосвязанных и согласованно функционирующих аппаратных и программных компонентов. Аппаратными компонентами локальной сети являются компьютеры и различное коммуникационное оборудование (кабельные системы, концентраторы и т. д.). Программными компонентами ВС являются операционные системы (ОС) и сетевые приложения.

Компоновкой сети называется процесс составления аппаратных компонентов с целью достижения нужного результата.

В зависимости от того, как распределены функции между компьютерами сети, они могут выступать в трех разных ролях:

1. Компьютер, занимающийся исключительно обслуживанием запросов других компьютеров, играет роль выделенного сервера сети (рис. 1.4).
2. Компьютер, обращающийся с запросами к ресурсам другой машины, играет роль узла-клиента (рис. 1.5).
3. Компьютер, совмещающий функции клиента и сервера, является одноранговым узлом (рис. 1.6).

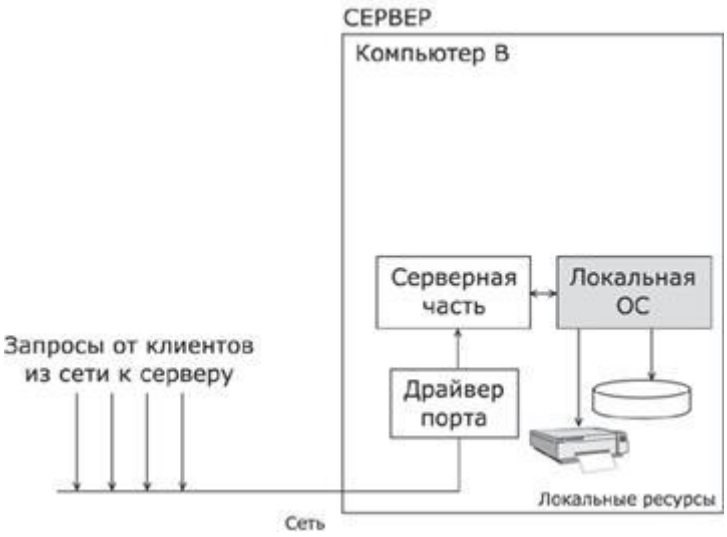


Рис. 1.4. Компьютер - выделенный сервер сети

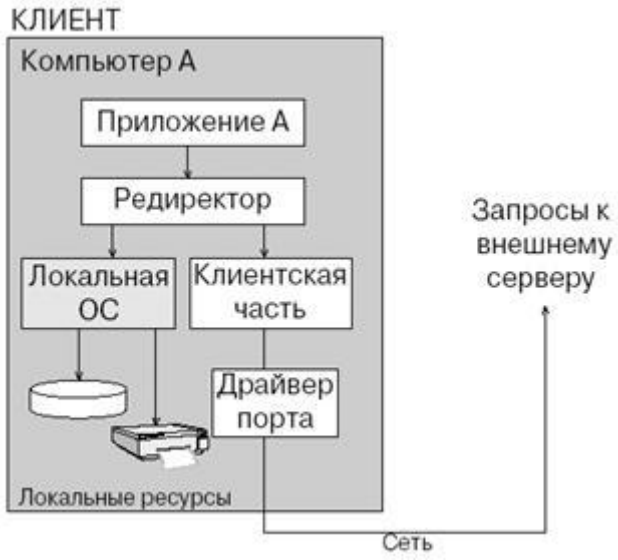


Рис. 1.5. Компьютер в роли узла-клиента

Очевидно, что сеть не может состоять только из клиентских или только из серверных узлов.

Сеть может быть построена по одной из трех схем:

- сеть на основе одноранговых узлов – одноранговая сеть;
- сеть на основе клиентов и серверов – сеть с выделенными серверами;
- сеть, включающая узлы всех типов – гибридная сеть.

Каждая из этих схем имеет свои достоинства и недостатки, определяющие их области применения.

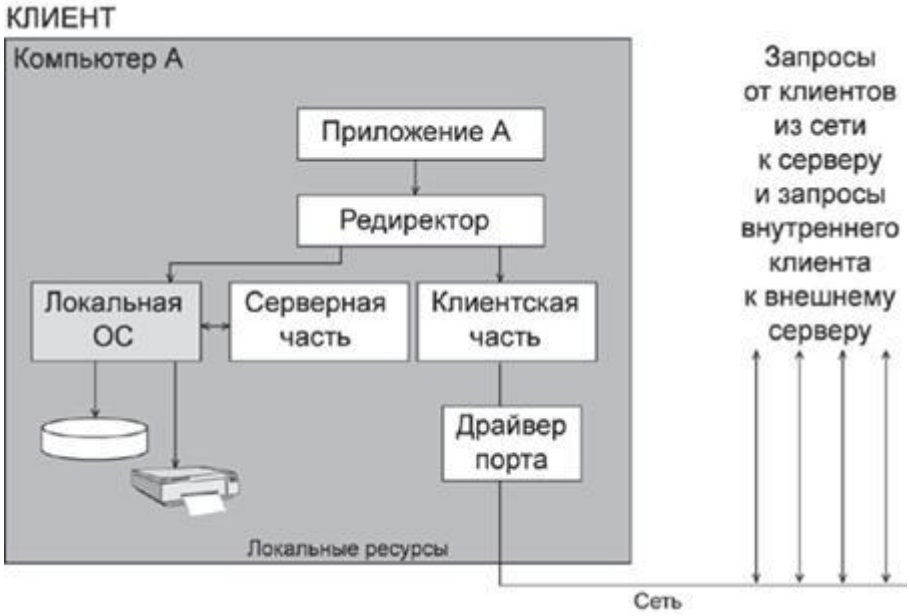


Рис. 1.6. Компьютер - одноранговый узел

В одноранговых сетях один и тот же ПК может быть и сервером, и клиентом, в том числе и клиентом своего клиента. В иерархических сетях разделяемые ресурсы хранятся только на сервере, сам сервер может быть клиентом только другого сервера более высокого уровня иерархии.

При этом каждый из серверов может быть реализован как на отдельном компьютере, так и в небольших по объему ЛВС, быть совмещенным на одном компьютере с каким-либо другим сервером.

Существуют и комбинированные сети, сочетающие лучшие качества одноранговых сетей и сетей на основе сервера. Многие администраторы считают, что такая сеть наиболее полно удовлетворяет их запросы.

Архитектура сети определяет основные элементы сети, характеризует ее общую логическую организацию, техническое обеспечение, программное обеспечение, описывает методы кодирования. Архитектура также определяет принципы функционирования и интерфейс пользователя.

Далее будет рассмотрено три вида архитектур:

- архитектура терминал-главный компьютер;
- одноранговая архитектура;
- архитектура клиент-сервер.

Архитектура терминал-главный компьютер

Архитектура терминал-главный компьютер (terminal-host computer architecture) – это концепция информационной сети, в которой вся обработка данных осуществляется одним или группой главных компьютеров.

Рассматриваемая архитектура предполагает два типа оборудования:

- главный компьютер, где осуществляется управление сетью, хранение и обработка данных;
- терминалы, предназначенные для передачи главному компьютеру команд на организацию сеансов и выполнения заданий, ввода данных для выполнения заданий и получения результатов.

Главный компьютер через МПД взаимодействуют с терминалами, как представлено на рис. 1.7.

Классический пример архитектуры сети с главными компьютерами – системная сетевая архитектура (System Network Architecture – SNA).

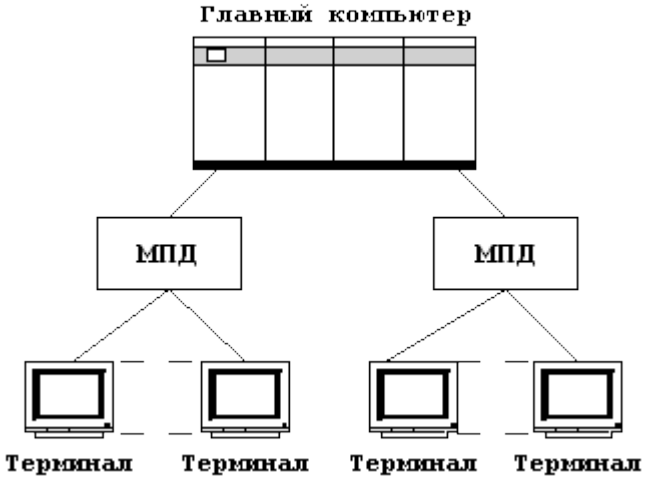


Рис. 1.7. Архитектура терминал-главный компьютер

Одноранговая архитектура

Одноранговая архитектура (peer-to-peer architecture) – это концепция информационной сети, в которой ее ресурсы рассредоточены по всем системам. Данная архитектура характеризуется тем, что в ней все системы равноправны.

К одноранговым сетям относятся малые сети, где любая рабочая станция может выполнять одновременно функции файлового сервера и рабочей станции. В одноранговых ЛВС дисковое пространство и файлы на любом компьютере могут быть общими. Чтобы ресурс стал общим, его необходимо отдать в общее пользование, используя службы удаленного доступа сетевых одноранговых операционных систем. В зависимости от того, как будет установлена защита данных, другие пользователи смогут пользоваться файлами сразу же после их создания. Одноранговые ЛВС достаточно хороши только для небольших рабочих групп.

Одноранговые ЛВС являются наиболее легким и дешевым типом сетей для установки. При соединении компьютеров, пользователи могут предоставлять ресурсы и информацию в совместное пользование.

Одноранговые сети имеют следующие преимущества:

- они легки в установке и настройке;
- отдельные ПК не зависят от выделенного сервера;
- пользователи в состоянии контролировать свои ресурсы;
- малая стоимость и легкая эксплуатация;
- минимум оборудования и программного обеспечения;
- нет необходимости в администраторе;
- хорошо подходят для сетей с количеством пользователей, не превышающим десяти.

Проблемой одноранговой архитектуры является ситуация, когда компьютеры отключаются от сети. В этих случаях из сети исчезают виды сервиса, которые они предоставляли. Сетевую безопасность одновременно можно применить только к одному ресурсу, и пользователь должен помнить столько паролей, сколько сетевых ресурсов. При получении доступа к разделяемому ресурсу ощущается падение производительности компьютера. Существенным недостатком одноранговых сетей является отсутствие централизованного администрирования.

Использование одноранговой архитектуры не исключает применения в той же сети также архитектуры терминал-главный компьютер или архитектуры клиент-сервер.

Архитектура клиент-сервер

Архитектура клиент-сервер (client-server architecture) – это концепция информационной сети, в которой основная часть ее ресурсов сосредоточена в серверах, обслуживающих своих клиентов (рис. 1.8). Рассматриваемая архитектура определяет два типа компонентов: серверы и клиенты.

Сервер – это объект, предоставляющий сервис другим объектам сети по их запросам. Сервис – это процесс обслуживания клиентов.

Сервер работает по заданиям клиентов и управляет выполнением их заданий. После выполнения каждого задания сервер посылает полученные результаты клиенту, пославшему это задание.

Сервисная функция в архитектуре клиент-сервер описывается комплексом прикладных программ, в соответствии с которым выполняются разнообразные прикладные процессы.

Рис. 1.8. Архитектура клиент – сервер

Процесс, который вызывает сервисную функцию с помощью определенных операций, называется клиентом. Им может быть программа или пользователь. На рис. 1.9 приведен перечень сервисов в архитектуре клиент-сервер.

Клиенты – это рабочие станции, которые используют ресурсы сервера и предоставляют удобные интерфейсы пользователя. Интерфейсы пользователя (рис. 1.9) это процедуры взаимодействия пользователя с системой или сетью.

В сетях с выделенным файловым сервером на выделенном автономном ПК устанавливается серверная сетевая операционная система. Этот ПК становится сервером. ПО,

установленное на рабочей станции, позволяет ей обмениваться данными с сервером. Наиболее распространенные сетевые операционная системы:

- NetWare фирмы Novel;
- Windows NT фирмы Microsoft;
- UNIX фирмы AT&T;
- Linux.

Помимо сетевой операционной системы необходимы сетевые прикладные программы, реализующие преимущества, предоставляемые сетью.

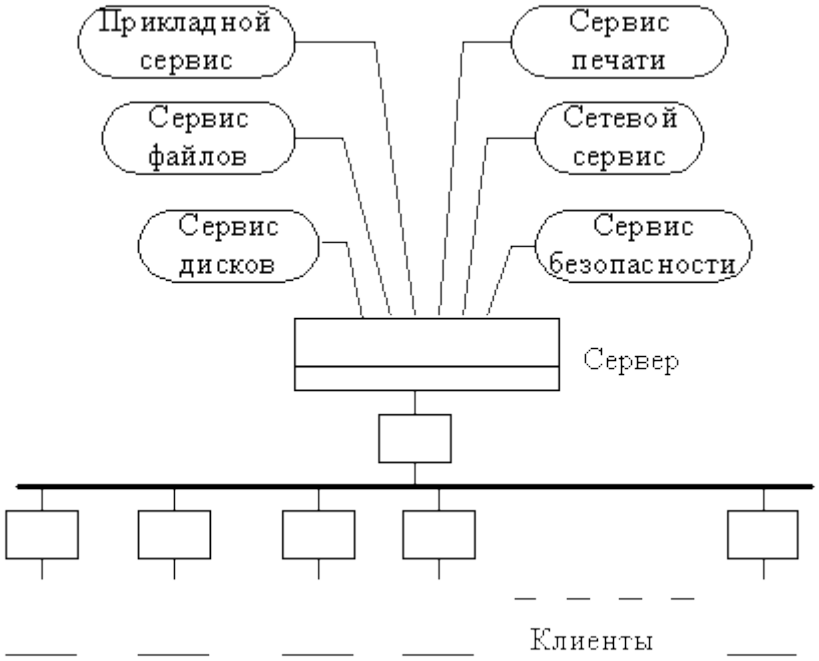


Рис. 1.9. Модель клиент-сервер

Круг задач, которые выполняют серверы в иерархических сетях, многообразен и сложен. Чтобы приспособиться к возрастающим потребностям пользователей, серверы в ЛВС стали специализированными. Так, например, в операционной системе Windows NT Server существуют различные типы серверов:

1. Файл-серверы и принт-серверы. Они управляют доступом пользователей к файлам и принтерам. Так, например, для работы с текстовым документом вы прежде всего запускаете на своем компьютере (PC) текстовыйпроцессор. Далее требуемый документ текстового процессора, хранящийся на файл-сервере, загружается в память PC, и таким образом Вы можете работать с этим документом на PC. Другими словами, файл-сервер предназначен для хранения файлов и данных.
2. Серверы приложений (в том числе сервер баз данных (БД), WEB-сервер). На них выполняются прикладные части клиент серверных приложений (программ). Эти серверы принципиально отличаются от файл-серверов тем, что при работе с файл-сервером нужный файл или данные целиком копируются на запрашивающий PC, а при работе с сервером приложений на PC пересылаются только результаты запроса. Например, по запросу можно получить только список работников, родившихся в сентябре, не загружая при этом в свою PC всю базу данных персонала.
3. Почтовые серверы управляют передачей электронных сообщений между пользователями сети.
4. Факс-серверы управляют потоком входящих и исходящих факсимильных сообщений через один или несколько факс-модемов.
5. Коммуникационные серверы управляют потоком данных и почтовых сообщений между данной ЛВС и другими сетями или удаленными пользователями через модем и телефонную линию. Они же обеспечивают доступ к Internet.
6. Сервер служб каталогов предназначен для поиска, хранения и защиты информации в сети. Windows NT Server объединяет PC в логические группы-домены, система

защиты которых наделяет пользователей различными правами доступа к любому сетевому ресурсу.

Клиент является инициатором и использует электронную почту или другие сервисы сервера. В этом процессе клиент запрашивает вид обслуживания, устанавливает сеанс, получает нужные ему результаты и сообщает об окончании работы.

Сети на базе серверов имеют лучшие характеристики и повышенную надежность. Сервер владеет главными ресурсами сети, к которым обращаются остальные рабочие станции.

В современной клиент-серверной архитектуре выделяется четыре группы объектов: клиенты, серверы, данные и сетевые службы. Клиенты располагаются в системах на рабочих местах пользователей. Данные в основном хранятся в серверах. Сетевые службы являются совместно используемыми серверами и данными. Кроме того службы управляют процедурами обработки данных.

Сети клиент-серверной архитектуры имеют следующие преимущества:

- позволяют организовывать сети с большим количеством рабочих станций;
- обеспечивают централизованное управление учетными записями пользователей, безопасностью и доступом, что упрощает сетевое администрирование;
- эффективный доступ к сетевым ресурсам;
- пользователю нужен один пароль для входа в сеть и для получения доступа ко всем ресурсам, на которые распространяются права пользователя.

Наряду с преимуществами сети клиент-серверной архитектуры имеют и ряд недостатков:

- неисправность сервера может сделать сеть неработоспособной;
- требуют квалифицированного персонала для администрирования;
- имеют более высокую стоимость сетей и сетевого оборудования.

Выбор архитектуры сети

Выбор архитектуры сети зависит от назначения сети, количества рабочих станций и от выполняемых на ней действий.

Следует выбрать одноранговую сеть, если:

- количество пользователей не превышает десяти;
- все машины находятся близко друг от друга;
- имеют место небольшие финансовые возможности;
- нет необходимости в специализированном сервере, таком как сервер БД, факс-сервер или какой-либо другой;
- нет возможности или необходимости в централизованном администрировании.

Следует выбрать клиент-серверную сеть, если:

- количество пользователей превышает десять;
- требуется централизованное управление, безопасность, управление ресурсами или резервное копирование;
- необходим специализированный сервер;
- нужен доступ к глобальной сети;
- требуется разделять ресурсы на уровне пользователей.

Лекция №15. Архитектура Ethernet

1.14.1 Вопросы лекции:

1. Эволюция Ethernet.
2. Спецификация Fast Ethernet.

1.14.2 Краткое содержание вопросов:

1. Эволюция Ethernet.

Ethernet, возникшая как сетевая технология с разделением среды передачи, а именно коаксиального кабеля, эволюционировала вместе с изменениями запросов пользователей.

Соответствуя самым последним требованиям к кабельной проводке, стандарт Ethernet распространяется теперь на такие среды передачи данных, как оптическое волокно и неэкранированная витая пара. Побудительными мотивами перехода к этим средам стало быстрое и всепроникающее распространение локальных сетей в коммерческих, правительственных и другого рода организациях, а также потребность в эффективном и экономичном управлении и обслуживании данных сетей. Прежние же неструктурированные проводки из коаксиального кабеля не удовлетворяли этим требованиям.

Коммутация в Ethernet была разработана с целью расширения доступной для серверов и рабочих станций полосы пропускания. Появление недорогих коммутаторов для локальных сетей предопределило переход к разреженным и частным (выделенным) сетям с помощью микросегментации.

В начале 1960 ряд исследователей, многие из которых в дальнейшем участвовали в проекте ARPANET, видели громадный потенциал возможности компьютеров обмениваться данными друг с другом. Установленное в 1965 году соединение между компьютерами в Массачусеттском Институте Технологии и Университете Южной Калифорнии явилось эмбрионом Интернета.

Интернет был рожден в 1969 году, когда компьютеры четырех университетских центров США были соединены между собой. До середины 70-х, когда имя Интернет вошло в обиход, он носил имя ARPANET. По сравнению с традиционными телефонными сетями, основанными на коммутации каналов, технологии ARPANET использовали коммутацию пакетов - данных ограниченного объема, заключенных в "конверты" с указанием источника и получателя. Эта технология позволила существенно упростить архитектуру сети и повысить ее надежность.

Электронная почта (e-mail) и telnet (удаленный доступ в режиме терминала) появились в ARPANET в 1972, а ftp (обмен файлами) - годом позже. В то же десятилетие была разработана архитектура TCP/IP, которая была окончательно внедрена в начале 1980-х.

В 1986 году основой сети являлась национальная опорная сеть США - NSFNET с пропускной способностью в 56К. Основные приложения Сети - e-mail, ftp и telnet, стали стандартными на компьютерах того времени, что послужило существенному увеличению числа пользователей, особенно в университетах и научных центрах.

В то время как e-mail и ftp позволяли пользователям поддерживать деловые и социальные контакты и обмениваться информацией, а telnet - использовать удаленные вычислительные ресурсы, разрабатывалось все больше приложений, позволяющих каталогизировать и находить информацию в Сети. Сегодня мало кто помнит приложения Archie, WAIS или gopher, которые явились предтечей сегодняшнего вэба и поисковых машин.

1989 год ознаменовал рождение нового протокола, который стал основой системы WorldWideWeb- http. В поисках возможности объединения распределенных информационных ресурсов - в основном, документов, хранящихся на различных компьютерах, - Тим Бернерс-Ли (TimBerners-Lee), в то время сотрудник CERN, предложил идею гиперлинков, через которые пользователь может переходить с одного документа на другой, находящийся, возможно, на другом компьютере и на другом континенте.

До начала 90-х Интернет в США в основном финансировался государством и его использование было доступно только для научно-образовательных учреждений. Подобная ситуация была и в Европе. С начала 90-х начинается коммерциализация Интернета и появляется все больше услуг, предлагаемых обычным пользователям. Соответственно, растет и число пользователей Интернета.

Де-регулирование и приватизация рынка телекоммуникаций в конце 1990-х, появление и широкое распространение персональных компьютеров и скоростных модемов открыло невиданные горизонты для развития Интернета. Рост пропускной способности стимулировал создание новых форм информационного контента, который в свою очередь требовал больших скоростей. Эта спираль инновации продолжается и сегодня, подстегнутая стремительным ростом мобильного Интернета и, как следствие, его размера и возможностей.

2. Спецификация FastEthernet.

Fast Ethernet — спецификация IEEE 802.3 u официально принятая 26 октября 1995 года определяет стандарт протокола канального уровня для сетей работающих при использовании как медного, так и волоконно-оптического кабеля со скоростью 100Мб/с. Новая спецификация является наследницей стандарта Ethernet IEEE 802.3, используя такой же формат кадра, механизм доступа к среде CSMA/CD и топологию звезда. Эволюция коснулась нескольких элементов конфигурации средств физического уровня, что позволило увеличить пропускную способность, включая типы применяемого кабеля, длину сегментов и количество концентраторов.

FastEthernet (Быстрый Ethernet) – высокоскоростная технология, предложенная фирмой 3Com для реализации сети Ethernet со скоростью передачи данных 100 Мбит/с, сохранившая в максимальной степени особенности 10-мегабитного Ethernet (Ethernet-10) и реализованная в виде стандарта 802.3u.

Основной целью при разработке технологии FastEthernet было обеспечение преемственности по отношению к 10-мегабитному Ethernet за счёт сохранения формата кадров и метода доступа CSMA/CD, что позволяет использовать прежнее программное обеспечение и средства управления сетями Ethernet. Одним из требований было также использование кабельной системы на основе витой пары категории 3, получившей на момент появления FastEthernet широкое распространение в сетях Ethernet-10. В связи с этим все отличия FastEthernet от Ethernet-10 сосредоточены на физическом уровне.

В FastEthernetпредусмотрены 3 варианта кабельных систем:

• многомодовый ВОК (используется 2 волокна);

• витая пара категории 5 (используется 2 пары);

• витая пара категории 3 (используется 4 пары).

Структура сети – иерархическая древовидная, построенная на концентраторах (как 10Base-T и 10Base-F), поскольку не предусматривалось использование коаксиального кабеля.

Диаметр сети Fast Ethernet, как показано в п.3.2.5, составляет немногим более 200 метров, что объясняется уменьшением времени передачи кадра минимальной длины в 10 раз в результате увеличения пропускной способности канала в 10 раз по сравнению с Ethernet-10. Тем не менее, возможно построение крупных сетей на основе технологии Fast Ethernet, благодаря появлению в начале 90-х годов прошлого века коммутаторов. При использовании коммутаторов протокол Fast Ethernet может работать в полнодуплексном режиме, в котором нет ограничений на общую длину сети, а остаются только ограничения на длину физических сегментов, соединяющих соседние устройства (адаптер – коммутатор или коммутатор – коммутатор).

Стандарт IEEE 802.3u определяет 3 спецификации физического уровня Fast Ethernet, несовместимых друг с другом:

• 100Base-TX – для передачи данных используются две неэкранированные пары UTP категории 5 или STP Type 1;

• 100Base-T4 – для передачи данных используются четыре неэкранированных пары UTP категорий 3, 4 или 5;

• 100Base-FX – для передачи данных используются два волокна многомодового ВОК.

Спецификации 100Base-TX и 100Base-FX.

Технологии 100Base-TX и 100Base-FX, несмотря на использование разных кабельных систем, имеют много общего с точки зрения построения и функционирования, в том числе, одинаковый метод логического кодирования – 4 B/5B при различных методах физического кодирования – MLT-3 в 100Base-TX и NRZI в 100Base-FX.

Кроме того, в технологии 100Base-TX имеется функция автопереговоров, обеспечивающая автоматическое определение скорости передачи (10 или 100 Мбит/с) между двумя связанными устройствами (СА, концентратор, коммутатор) путем посылки при подключении пачки специальных импульсов FLP – Fast Link Pulse burst – со стороны устройства, которое может работать на скорости 100 Мбит/с. Если встречное устройство не откликается на эти импульсы, это означает, что оно может работать только на скорости 10 Мбит/с, и первое устройство устанавливает режим передачи данных 10 Мбит/с.

Спецификация 100Base-T4.

К моменту появления Fast Ethernet большинство ЛВС Ethernet в качестве кабельной системы использовали неэкранированную витую пару категории 3. Желание сохранить кабельную систему 10-мегабитных ЛВС Ethernet обусловило применение специального метода логического кодирования – 8 B/6T, обеспечившего более узкий спектр сигнала, что при скорости 33 Мбит/с позволило уложиться в полосу 16 МГц витой пары категории 3.

При кодировании 8B/6T 8 бит заменяются 6-ю троичными цифрами. Длительность одной троичной цифры – 40 нс. Следовательно, один байт передается за 240 нс (6*40 нс), что соответствует скорости передачи в 33,3 Мбит/с. Для передачи данных используется 3 пары UTP категории 3 (3*33,3 Мбит/с = 100 Мбит/с), и еще одна пара используется для прослушивания несущей с целью обнаружения коллизий.

Скорость изменения сигнала на каждой паре составляет: 1/(40 нс) = 25 Мбод, что позволяет использовать витую пару категории 3/

Чтобы лучше понять работу и разобраться во взаимодействии элементов Fast Ethernet обратимся к рисунку 29.

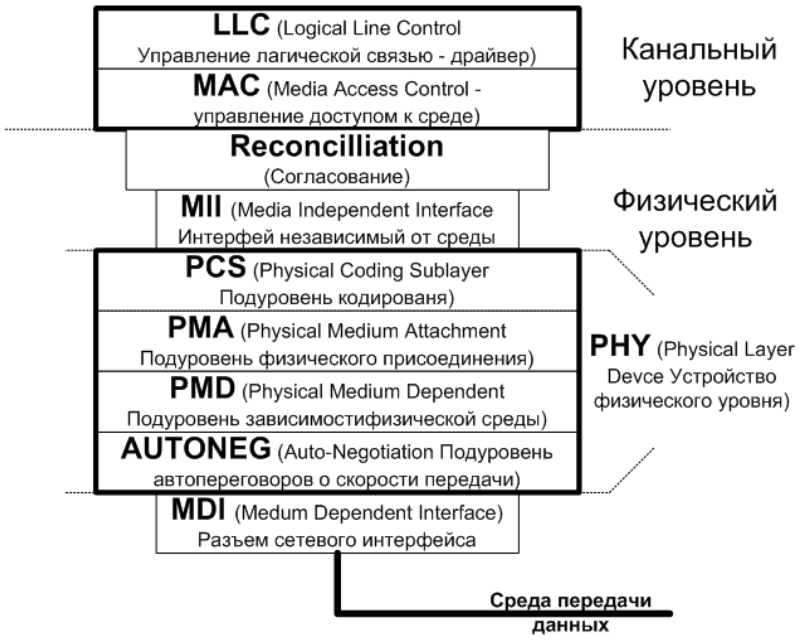


Рисунок 29. Система Fast Ethernet

В спецификации IEEE 802.3 u функции канального уровня разбиты на два подуровня: управления логической связью (LLC) и уровня доступа к среде (MAC), который будет рассмотрен ниже. LLC, функции которого определены стандартом IEEE 802.2, фактически обеспечивает взаимосвязь с протоколами более высокого уровня, (например, с IP или IPX), предоставляя различные коммуникационные услуги:

- **Сервис без установления соединения и подтверждений приема.** Простой сервис, который не обеспечивает управления потоком данных или контроля ошибок, а также не гарантирует правильную доставку данных.
- **Сервис с установлением соединения.** Абсолютно надежный сервис, который гарантирует правильную доставку данных за счет установления соединения с системой-приемником до начала передачи данных и использования механизмов контроля ошибок и управления потоком данных.
- **Сервис без установления соединения с подтверждениями приема.** Средний по сложности сервис, который использует сообщения подтверждения приема для обеспечения гарантированной доставки, но не устанавливает соединения до передачи данных.

На передающей системе данные, переданные вниз от протокола Сетевого уровня, вначале инкапсулируются подуровнем LLC. Стандарт называет их Protocol Data Unit (PDU,

протокольный блок данных). Когда PDU передается вниз подуровню MAC, где снова обрамляется заголовком и постинформацией, с этого момента технически его можно назвать кадром. Для пакета Ethernet это означает, что кадр 802.3 помимо данных Сетевого уровня содержит трехбайтовый заголовок LLC. Таким образом, максимально допустимая длина данных в каждом пакете уменьшается с 1500 до 1497 байтов.

Заголовок LLC состоит из трех полей:

- **DSAP** (1 байт) *Destination Service Access Point* — точка доступа к сервису системы — получателя указывает, в каком месте буферов памяти системы-получателя следует разместить данные пакета.
- **SSAP** (1 байт) *Source Service Access Point* — точка доступа к сервису системы — источника выполняет такие же функции для источника данных, размещенных в пакете, на передающей системе.
- **Поле управления** (1 или 2 байта) указывает на тип сервиса, необходимого для данных в PDU и функций пакета. В зависимости от того, какой сервис нужно предоставить, поле управления может быть длиной 1 или 2 байта.

Принимающей системе необходимо определить, какой из протоколов Сетевого уровня должен получить входящие данные. В пакетах 802.3 в рамках PDU LLC применяется еще один протокол, называемый *Sub -Network Access Protocol (SNAP, протокол доступа к подсетям).*

Заголовок SNAP имеет длину 5 байт и располагается непосредственно после заголовка LLC в поле данных кадра 802.3, как показано на рисунке. Заголовок содержит два поля.

Код организации. Идентификатор организации или производителя — это 3-байтовое поле, которое принимает такое же значение, как первые 3 байта MAC-адреса отправителя в заголовке 802.3.

Локальный код. Локальный код — это поле длиной 2 байта, которое функционально эквивалентно полю Ethertype в заголовке Ethernet II.

Подуровень согласования

Как было сказано ранее Fast Ethernet это эволюционировавший стандарт. MAC рассчитанный на интерфейс AUI, необходимо преобразовать для интерфейса MII, используемого в Fast Ethernet, для чего и предназначен этот подуровень.

Управление доступом к среде (MAC)

Каждый узел в сети Fast Ethernet имеет контроллер доступа к среде **(Media AccessController** — MAC). MAC имеет ключевое значение в Fast Ethernet и имеет три назначения:

- определяет, когда узел может передать пакет;
- пересылает кадры уровню PHY для преобразования в пакеты и передачи в среду;
- получает кадры из уровня PHY и передает обрабатывающему их программному обеспечению (протоколам и приложениям).*

Самым важным из трех назначений MAC является первое. Для любой сетевой технологии, которая использует общую среду, правила доступа к среде, определяющие, когда узел может передавать, являются ее основной характеристикой. Разработкой правил доступа к среде занимаются несколько комитетов IEEE. Комитет 802.3, часто именуемый комитетом Ethernet, определяет стандарты на ЛВС, в которых используются правила под названием **CSMA/ CD** (Carrier Sense Multiple Access with Collision Detection — множественный доступ с контролем несущей и обнаружением конфликтов).

CSMS/ CD являются правилами доступа к среде как для Ethernet, так и для Fast Ethernet. Именно в этой области две технологии полностью совпадают.

Поскольку все узлы в Fast Ethernet совместно используют одну и ту же среду, передавать они могут лишь тогда, когда наступает их очередь. Определяют эту очередь правила CSMA/ CD.

CSMA/ CD

Контроллер MAC Fast Ethernet, прежде чем приступить к передаче, прослушивает несущую. Несущая существует лишь тогда, когда другой узел ведет передачу. Уровень PHY

определяет наличие несущей и генерирует сообщение для MAC. Наличие несущей говорит о том, что среда занята и слушающий узел (или узлы) должны уступить передающему.

Устройство физического уровня (PHY)

Поскольку Fast Ethernet может использовать различный тип кабеля, то для каждой среды требуется уникальное предварительное преобразование сигнала. Преобразование также требуется для эффективной передачи данных: сделать передаваемый код устойчивым к помехам, возможным потерям, либо искажениям отдельных его элементов (бодов), для обеспечения эффективной синхронизации тактовых генераторов на передающей или приемной стороне.

Подуровень кодирования (PCS)

Кодирует/декодирует данные поступающие от/к уровня MAC с использованием алгоритмов 4B/5B или8B/6T.

Подуровни физического присоединения и зависимости от физической среды (PMA и PMD)

Подуровни PMA и PMD осуществляют связь между подуровнем PSC и интерфейсом MDI, обеспечивая формирование в соответствии с методом физического кодирования: NRZI или MLT-3.

Подуровень автопереговоров (AUTONEG)

Подуровень автопереговоров позволяет двум взаимодействующим портам автоматически выбирать наиболее эффективный режим работы: дуплексный или полудуплексный 10 или 100 Мб/с.

1.15 Лекция №17. Технология TokenRing.

1.15.1 Вопросы лекции:

1. Оборудование TR.
2. Топология TR.

1.15.2 Краткое содержание вопросов:

В настоящее время тенденция в развитии микропроцессоров и систем, построенных на их основе, направлена на все большее повышение их производительности. Вычислительные возможности любой системы достигают своей наивысшей производительности благодаря двум факторам:

использованию высокоскоростных элементов и параллельному выполнению большого числа операций. Направления, связанные с повышением производительности отдельных микропроцессоров, мы рассматривали в предыдущих лекциях, а в этой лекции остановимся на вопросах распараллеливания обработки информации.

Существует несколько вариантов классификации систем параллельной обработки данных. По-видимому, самой ранней и наиболее известной является классификация архитектур вычислительных систем, предложенная в 1966 году М. Флинном. Классификация базируется на понятии потока, под которым понимается последовательность элементов, команд или данных, обрабатываемая процессором. На основе числа потоков команд и потоков данных выделяются четыре класса архитектур:

SISD, MISD, SIMD, MIMD.

SISD (sINgle INsTRuction sTReam / sINgle data sTReam) - одиночный поток команд и одиночный поток данных. К этому классу относятся прежде всего классические последовательные машины, или, иначе, машины фон-неймановского типа. В таких машинах есть только один поток команд, все команды обрабатываются последовательно друг за другом и каждая команда инициирует одну операцию с одним потоком данных. Не имеет значения тот факт, что для увеличения скорости обработки команд и скорости выполнения арифметических операций процессор может использовать конвейерную обработку. В таком понимании машины данного класса фактически не относятся к параллельным системам.

SIMD (sINgle INsTRuction sTReam / multIPle data sTReam) - одиночный поток команд и множественный поток данных. Применительно к одному микропроцессору этот подход реализован в MMX- и SSE- расширениях современных микропроцессоров.Микропроцессорные системы типа SIMD состоят из большого числа идентичных процессорных элементов, имеющих собственную память. Все процессорные элементы в такой машине выполняют одну и ту же программу. Это позволяет выполнять одну арифметическую операцию сразу над многими данными - элементамивектора. Очевидно, что такая система, составленная из большого числа процессоров, может обеспечить существенное повышение производительности только на тех задачах, при решении которых все процессоры могут делать одну и ту же работу.

MISD (multIPe INsTRuction sTReam / sINgle data sTReam) - множественный поток команд и одиночный поток данных. Определениеподразумевает наличие в архитектуре многих процессоров, обрабатывающих один и тот же поток данных. Ряд исследователей к данному классу относят конвейерные машины.

MIMD (multIPe INsTRuction sTReam / multIPle data sTReam) - множественный поток команд и множественный поток данных. Базовоймоделью вычислений в этом случае является совокупность независимых процессов, эпизодически обращающихся к разделяемым данным. В такой системе каждый процессорный элемент выполняет свою программу достаточно независимо от других процессорных элементов.Архитектура MIMD дает большую гибкость: при наличии адекватной поддержки со стороны аппаратных средств и программного обеспечения MIMD может работать как однопользовательская система, обеспечивая высокопроизводительную обработку данных для одной прикладной задачи, как многопрограммная машина, выполняющая множество задач параллельно, и как некоторая комбинация этих возможностей. К тому же архитектура MIMD может использовать все преимущества современной микропроцессорной технологии на основе строгого учета соотношения стоимость/производительность. В действительности практически все современные многопроцессорные системы строятся на тех же микропроцессорах, которые можно найти в персональных компьютерах, рабочих станциях и небольших однопроцессорных серверах.

Как и любая другая, приведенная выше классификация несовершенна: существуют машины, прямо в нее не попадающие, имеются также важные признаки, которые в этой классификации не учтены. Рассмотрим классификацию многопроцессорных и многомашинных систем на основе другого признака - степени разделения вычислительных ресурсов системы.

В этом случае выделяют следующие 4 класса систем:

системы с симметричной мультипроцессорной обработкой (symmeTRic multIProcessINg), или SMP-системы;

системы, построенные по технологии неоднородного доступа к памяти (non-un IForm memory access), или NUMA-системы;

кластеры;

системы вычислений с массовым параллелизмом (massively parallel processor), или MPP-системы.

Самым высоким уровнем интеграции ресурсов обладает система с симметричной мультипроцессорной обработкой, или SMP-система (рис. 13.1).

В этой архитектуре все процессоры имеют равноправный доступ ко всему пространству оперативной памяти и ввода/вывода. Поэтому SMP-архитектура называется симметричной. Ее интерфейсы доступа к пространству ввода/вывода и ОП, система управления кэш-памятью,системное ПО и т. п. построены таким образом, чтобы обеспечить согласованный доступ к разделяемым ресурсам. Соответствующиемеханизмы блокировки заложены и в шинном интерфейсе, и в компонентах операционной системы, и при построении кэша.

Рис. 13.1. Система с симметричной мультипроцессорной обработкой

С точки зрения прикладной задачи, SMP-система представляет собой единый вычислительный комплекс с вычислительными ресурсами, пропорциональными количеству процессоров. Распараллеливание вычислений обеспечивается операционной системой, установленной на одном из процессоров. Вся система работает под управлением единой ОС (обычно UNIX-подобной, но для Intel-платформ поддерживается WINdows NT).

ОС автоматически в процессе работы распределяет процессы по процессорным ядрам, оптимизируя использование ресурсов. Ядра задействуются равномерно, и прикладные программы могут выполняться параллельно на всем множестве ядер. При этом достигается максимальное быстродействие системы. Важно, что для синхронизации приложений вместо сложных механизмов и протоколовмежпроцессорной коммуникации применяются стандартные функции ОС. Таким образом, проще реализовать проекты с распараллеливанием программных потоков. Общая для совокупности ядер ОС позволяет с помощью служебных инструментов собирать статистику, единую для всей архитектуры. Соответственно, можно облегчить отладку и оптимизацию приложений на этапе разработки или масштабирования для других форм многопроцессорной обработки.

В общем случае приложение, написанное для однопроцессорной системы, не требует модификации при его переносе в мультипроцессорную среду. Однако для оптимальной работы программы или частей ОС они переписываются специально для работы в мультипроцессорной среде.

Сравнительно небольшое количество процессоров в таких машинах позволяет иметь одну централизованную общую память и объединить процессоры и память с помощью одной шины.

Сдерживающим фактором в подобных системах является пропускная способность магистрали, что приводит к их плохой масштабируемости. Причиной этого является то, что в каждый момент времени шина способна обрабатывать только одну транзакцию, вследствие чего возникают проблемы разрешения конфликтов при одновременном обращении нескольких процессоров к одним и тем же областям общей физической памяти. Вычислительные элементы начинают мешать друг другу. Когда произойдет такой конфликт, зависит от скорости связи и от количества вычислительных элементов. Кроме того, системная шина имеет ограниченное число слотов. Все это очевидно препятствуетувеличению производительности при увеличении числа процессоров. В реальных системах можно задействовать не более 32 процессоров.

В современных микропроцессорах поддержка построения мультипроцессорной системы закладывается на уровне аппаратной реализации МП, что делает многопроцессорные системы сравнительно недорогими.

Так, для обеспечения возможности работы на общую магистраль каждый микропроцессор фирмы Intel начиная с Pentium Pro имеет встроенную поддержку двухразрядного идентификатора процессора - APIC (Advanced Programmable INTerrupt ConTRoller). По умолчанию CPUс самым высоким номером идентификатора становится процессором начальной загрузки.
Такая идентификация облегчает арбитраж шины данных в SMP-системе. Подобные средства мы видели и в МП Power4, где на аппаратном уровне поддерживается создание микросхемного модуля MCM из 4 микропроцессоров, включающего в совокупности 8 процессорных ядер.

Сегодня SMP широко применяют в многопроцессорных суперкомпьютерах и серверных приложениях. Однако если необходимо детерминированное исполнение программ в реальном масштабе времени, например, при визуализации мультимедийных данных, возможности сугубосимметричной обработки весьма ограничены. Может возникнуть ситуация, когда приложения, выполняемые на различных ядрах, обращаются к одному ресурсу ОС. В этом случае доступ получит только одно из ядер.

Остальные будут простаивать до высвобождения критической области.

Естественно, при этом резко снижается производительность приложений реального времени.

Исчерпание производительности системной шины в SMP-системах при доступе большого числа процессоров к общему пространству оперативной памяти и принципиальные ограничения шинной технологии стали причиной сдерживания роста производительности SMP-систем. На данный момент эта проблема получила два решения. Первое - замена системной шины на высокопроизводительный коммутатор, обеспечивающий одновременный неблокирующий доступ к различным участкам памяти. Второе решение предлагает технология NUMA.

Система, построенная по технологии NUMA, представляет собой набор узлов, каждый из которых, по сути, является функционально законченным однопроцессорным или SMP-компьютером. Каждый имеет свое локальное пространство оперативной памяти и ввода/вывода. Но с помощью специальной логики каждый имеет доступ к пространству оперативной памяти и ввода/вывода любого другого узла (рис. 13.2). Физически отдельные устройства памяти могут адресоваться как логически единое адресное пространство - это означает, что любойпроцессор может выполнять обращения к любым ячейкам памяти, в предположении, что он имеет соответствующие права доступа. Поэтому иногда такие системы называются системами с распределенной разделяемой памятью (DSM - disTRibuted shared memory).

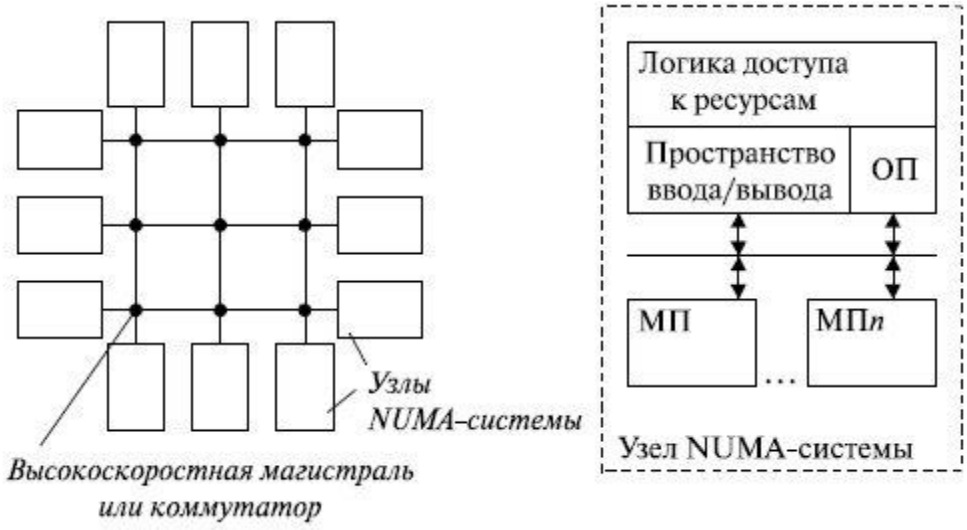


Рис. 13.2. Система, построенная по технологии неоднородного до ступа к памяти

При такой организации память каждого узла системы имеет свою адресацию в адресном пространстве всей системы. Логика доступа к ресурсам определяет, к памяти какого узла относится выработанный процессором адрес. Если он не принадлежит памяти данного узла, организуется обращение к другому узлу согласно заложенной в логике доступа карте адресов. При этом доступ к локальной памяти осуществляется в несколько раз быстрее, чем к удаленной.

При использовании наиболее распространенного сейчас варианта cc-NUMA (cache-coherent NUMA - неоднородный доступ к памяти с согласованием содержимого кэш-памяти) обеспечивается кэширование данных оперативной памяти других узлов.

Обычно вся система работает под управлением единой ОС, как в SMP. Возможны также варианты динамического разделения системы, когда отдельные разделы системы работают под управлением разных ОС.

Довольно большое время доступа к оперативной памяти соседних узлов по сравнению с доступом к ОП своего узла в NUMA-системах на настоящий момент делает такое использование не вполне оптимальным.

Так что полной функциональностью SMP-систем NUMA-компьютеры на сегодняшний день не обладают. Однако среди систем общего назначения NUMA-системы имеют один из наиболее высоких показателей по масштабируемости и,
соответственно, попроизводительности. На сегодня максимальное число процессоров в cc-NUMA-системах может превышать 1000 (серия OrigIN3000). Один из наиболее производительных суперкомпьютеров - Tera 10 - имеет производительностью 60 Тфлопс и состоит из 544 SMP-узлов, в каждом из которых находится от 8 до 16 процессоров Itanium 2.

Следующим уровнем в иерархии параллельных систем являются комплексы, также состоящие из отдельных машин, но лишь частично разделяющие некоторые ресурсы. Речь идет о кластерах.

1.16 Лекция №17. Технология TokenRing.

1.16.1 Вопросы лекции:

1. Оборудование TR.
2. Топология TR.

1.16.2 Краткое содержание вопросов:

1. Оборудование TR.

Сети Token Ring, так же как и сети Ethernet, характеризует разделяемая среда передачи данных, которая в данном случае состоит из отрезков кабеля, соединяющих все станции сети в кольцо. Кольцо рассматривается как общий разделяемый ресурс, и для доступа к нему требуется не случайный алгоритм, как в сетях Ethernet, а детерминированный, основанный на передаче станциям права на использование кольца в определенном порядке. Это право передается с помощью кадра специального формата, называемого *маркером* или *токеном (token)*.

Технология Token Ring был разработана компанией IBM в 1984 году, а затем передана в качестве проекта стандарта в комитет IEEE 802, который на ее основе принял в 1985 году стандарт 802.5. Компания IBM использует технологию Token Ring в качестве своей основной сетевой технологии для построения локальных сетей на основе компьютеров различных классов - мэйнфреймов, мини-компьютеров и персональных компьютеров. В настоящее время именно компания IBM является основным законодателем моды технологии Token Ring, производя около 60 % сетевых адаптеров этой технологии.

Сети Token Ring работают с двумя битовыми скоростями - 4 и 16 Мбит/с. Смешение станций, работающих на различных скоростях, в одном кольце не допускается. Сети Token

Ring, работающие со скоростью 16 Мбит/с, имеют некоторые усовершенствования в алгоритме доступа по сравнению со стандартом 4 Мбит/с.

Технология Token Ring является более сложной технологией, чем Ethernet. Она обладает свойствами отказоустойчивости. В сети Token Ring определены процедуры контроля работы сети, которые используют обратную связь кольцеобразной структуры - посланный кадр всегда возвращается в станцию - отправитель. В некоторых случаях обнаруженные ошибки в работе сети устраняются автоматически, например может быть восстановлен потерянный маркер. В других случаях ошибки только фиксируются, а их устранение выполняется вручную обслуживающим персоналом.

Для контроля сети одна из станций выполняет роль так называемого *активного монитора*. Активный монитор выбирается во время инициализации кольца как станция с максимальным значением MAC-адреса, Если активный монитор выходит из строя, процедура инициализации кольца повторяется и выбирается новый активный монитор. Чтобы сеть могла обнаружить отказ активного монитора, последний в работоспособном состоянии каждые 3 секунды генерирует специальный кадр своего присутствия. Если этот кадр не появляется в сети более 7 секунд, то остальные станции сети начинают процедуру выборов нового активного монитора.

Оборудование TR.

Концентратор TokenRing (MSAU) представляет собой набор блоков TCU (TrunkCouplingUnit – блок подключения к магистрали), к которым отдельными радиальными кабелями (lobecabling) подключаются станции. Блок TCU содержит реле, в нормальном состоянии замыкающее магистраль в обход порта (одновременно замыкает вход и выход порта со стороны станции). Если к порту подключена станция, то она выдает “фантомный” сигнал постоянного тока, переключающий реле. Если станция отключается от кольца или происходит обрыв кабеля, реле восстанавливает обходной путь. Станция, физически подключенная к TCU, может проверить свою линию до MSAU (поскольку TCU обеспечивает замыкание ее приемника на ее передатчик), и, в случае исправности линии, выдать “фантомный сигнал”.

Конструктивно концентратор представляет собой автономный блок с десятью разъемами на передней панели (рис. 40).

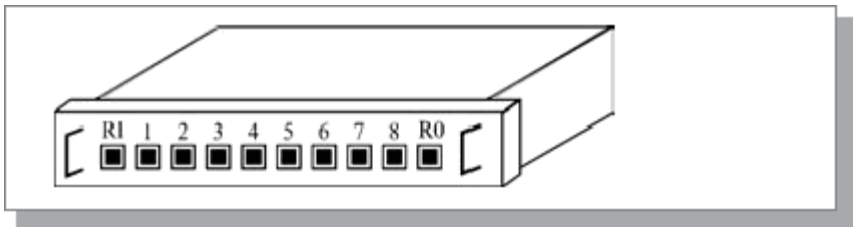


Рисунок 40. Концентратор Token-Ring (8228 MAU)

Кроме блоков TCU (обычно от 8 до 24), концентраторы MSAU имеют два порта для образования кольца концентраторов: порт RI (RingIn, вход кольца) и порт RO (RingOut, выход кольца). Эти порты также снабжены реле, обеспечивающим замыкание магистрали в обход отключенного порта.

Концентраторы могут быть пассивными и активными. ПассивныйMSAU обеспечивает только электрическое подключение станции к магистрали. Активный MASU имеет в каждом блоке TCU повторитель, восстанавливающий форму сигнала. Активные концентраторы могут содержать блок управления по SNMP или RMON. Сегментирующие (Portswitch) концентраторы позволяют организовывать несколько колец на одном устройстве.

Сетевые адаптеры содержат блок повторения, который может регенерировать сигнал и восстанавливать его синхронизацию (этим занимается только активный монитор). Для ресинхронизации используется 30-битный буфер, в котором накапливаются сигналы. Этот буфер подключается активным монитором к кольцу, и все данные пропускаются через него,

выходя с нужной частотой. (При максимальном количестве станций (260) смещение бита за оборот по кольцу может достигать трех битовых интервалов.)

Технология TokenRing позволяет использовать для магистральных и радиальных кабелей витую пару (UTP или STP) или оптоволокно. Расстояние между пассивными концентраторами может достигать 100 м (STPType 1) и 45 м (UTPCategory 3), а между активными – 730 м и 365 м соответственно. Использование оптоволокна увеличивает максимальную длину каждого сегмента до 1 км. Разные производители оборудования и программного обеспечения определяют различные ограничения, так что при проектировании сети TokenRing необходимо пользоваться данными выбранного производителя.

2. Топология TR.

Сеть ***Token-Ring*** имеет топологию кольцо, хотя внешне она больше напоминает звезду. Это связано с тем, что отдельные абоненты (компьютеры) присоединяются к сети не напрямую, а через специальные концентраторы или многостанционные устройства доступа (*MSAU*или *MAU* – Multistation *Access Unit*). Физически сеть образует звездно-кольцевую топологию (рисунок 41). В действительности же абоненты объединяются все-таки в кольцо, то есть каждый из них передает информацию одному соседнему абоненту, а принимает информацию от другого.

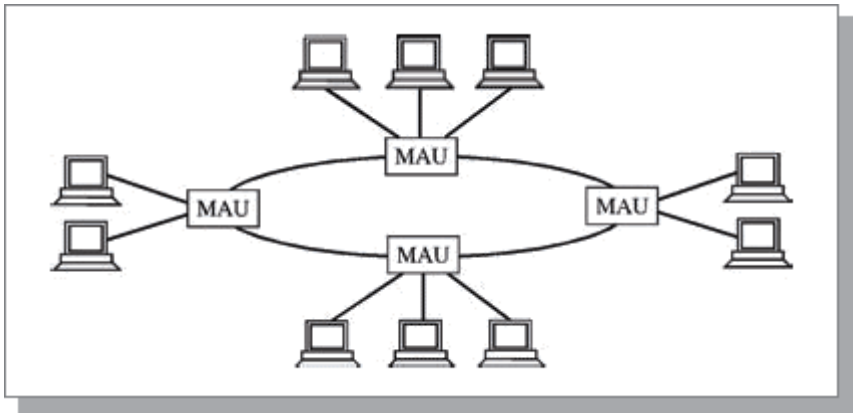


Рисунок 41. Звездно-кольцевая топология сети Token-Ring

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

Для каждого абонента в составе концентратора применяется специальный блок подключения к магистрали (TCU – *Trunk Coupling* Unit), который обеспечивает автоматическое включение абонента в кольцо, если он подключен к концентратору и исправен. Если *абонент*отключается от концентратора или же он неисправен, то блок TCU автоматически восстанавливает *целостность* кольца без участия данного абонента. Срабатывает TCU *по* сигналу постоянного тока (так называемый "фантомный" ток), который приходит от абонента, желающего включиться в кольцо. *Абонент* может также отключиться от кольца и провести процедуру *самотестирования*. "Фантомный" ток никак не влияет на информационный сигнал, так как сигнал в кольце не имеет постоянной составляющей.

Эта топология основана на топологии "физическое кольцо с подключением типа звезда". В данной топологии все рабочие станции подключаются к центральному концентратору (TokenRing) как в топологии физическая звезда. Центральный концентратор - это интеллектуальное устройство, которое с помощью перемычек обеспечивает последовательное соединение выхода одной станции со входом другой станции.

Другими словами с помощью концентратора каждая станция соединяется только с двумя другими станциями (предыдущей и последующей станциями). Таким образом, рабочие станции связаны петлей кабеля, по которой пакеты данных передаются от одной станции к

другой и каждая станция ретранслирует эти посланные пакеты. В каждой рабочей станции имеется для этого приемо-передающее устройство, которое позволяет управлять прохождением данных в сети. Физически такая сеть построена по типу топологии “звезда”.

Концентратор создаёт первичное (основное) и резервное кольца. Если в основном кольце произойдёт обрыв, то его можно обойти, воспользовавшись резервным кольцом, так как используется четырёхжильный кабель. Отказ станции или обрыв линии связи рабочей станции не вличет за собой отказ сети как в топологии кольцо, потому что концентратор отключет неисправную станцию и замкнет кольцо передачи данных.

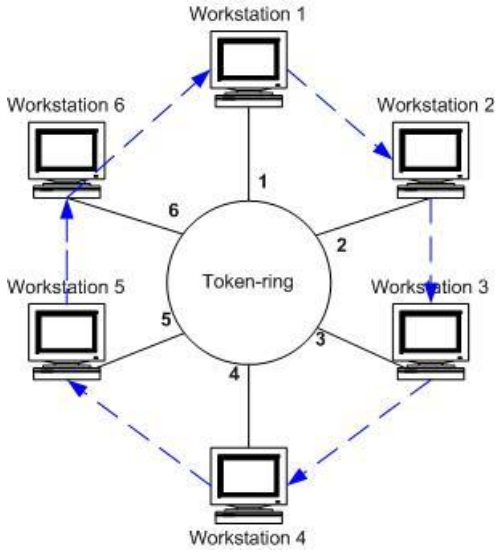


Рисунок 42 - Топология TokenRing

В архитектуре TokenRing маркер передаётся от узла к узлу по логическому кольцу, созданному центральным концентратором. Такая маркерная передача осуществляется в фиксированном направлении (направление движения маркера и пакетов данных представлено на рисунке стрелками синего цвета). Станция, обладающая маркером, может отправить данные другой станции.

Для передачи данных рабочие станции должны сначала дождаться прихода свободного маркера. В маркере содержится адрес станции, пославшей этот маркер, а также адрес той станции, которой он предназначается. После этого отправитель передает маркер следующей в сети станции для того, чтобы и та могла отправить свои данные.

Один из узлов сети (обычно для этого используется файл-сервер) создаёт маркер, который отправляется в кольцо сети. Такой узел выступает в качестве активного монитора, который следит за тем, чтобы маркер не был утерян или разрушен.

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

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

Все приведенные характеристики относятся к случаю использования неэкранированной витой пары. Если применяется другая *среда передачи*, характеристики сети могут отличаться. Например, при использовании экранированной витой пары (*STP*) количество абонентов может быть увеличено до 260 (вместо 96), *длина* кабеля – до 100 метров (вместо 45), количество концентраторов – до 33, а полная *длина*кольца, соединяющего *концентраторы* – до 200 метров. Оптоволоконный *кабель* позволяет увеличивать длину кабеля до двух километров.

Преимущества сетей топологии TokenRing:

- топология обеспечивает равный доступ ко всем рабочим станциям;

- высокая надежность, так как сеть устойчива к неисправностям отдельных станций и к разрывам соединения отдельных станций.

Недостатки сетей топологии TokenRing: большой расход кабеля и соответственно дорогостоящая разводка линий связи.

1.17 Лекция №18. Архитектура FDDI.

1.17.1 Вопросы лекции:

1. Изучить технологию FDDI;
2. Ознакомиться с особенностями метода доступа FDDI.

1.17.2 Краткое содержание вопросов:

Архитектура FDDI

Технология *FDDI (Fiber Distributed Data Interface)*- оптоволоконный интерфейс распределенных данных - это первая технология локальных сетей, в которой средой передачи данных является волоконно-оптический кабель.

Технология FDDI во многом основывается на технологии Token Ring, развивая и совершенствуя ее основные идеи.

Сеть FDDI строится на основе двух оптоволоконных колец, которые образуют основной и резервный пути передачи данных между узлами сети. Наличие двух колец - это основной способ повышения отказоустойчивости в сети FDDI, и узлы, которые хотят воспользоваться этим повышенным потенциалом надежности, должны быть подключены к обоим кольцам.

В нормальном режиме работы сети данные проходят через все узлы и все участки кабеля только первичного (Primary) кольца, этот режим назван режимом *Thru* - «сквозным» или «транзитным». Вторичное кольцо (Secondary) в этом режиме не используется.

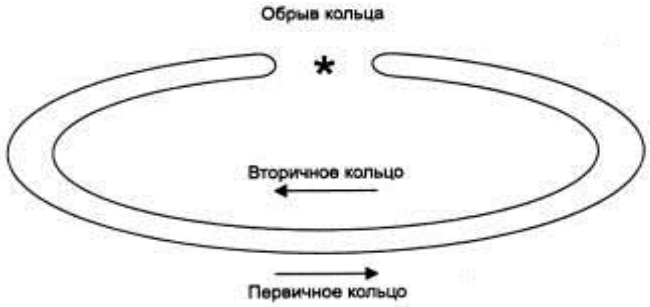


Рисунок 1 - Реконфигурация колец FDDI при отказе

В случае какого-либо вида отказа, когда часть первичного кольца не может передавать данные (например, обрыв кабеля или отказ узла), первичное кольцо объединяется со вторичным (рисунок 1), вновь образуя единое кольцо. Этот режим работы сети называется *Wrap,* то есть «свертывание» или «сворачивание» колец. Операция свертывания производится средствами концентраторов и/или сетевых адаптеров FDDI. Для упрощения этой процедуры данные по первичному кольцу всегда передаются в одном направлении (на диаграммах это направление изображается против часовой стрелки), а по вторичному - в обратном (изображается по часовой стрелке). Поэтому при образовании общего кольца из двух колец передатчики станций по-прежнему остаются подключенными к приемникам соседних станций, что позволяет правильно передавать и принимать информацию соседними станциями.

Технология FDDI дополняет механизмы обнаружения отказов технологии Token Ring механизмами реконфигурации пути передачи данных в сети, основанными на наличии резервных связей, обеспечиваемых вторым кольцом.

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

Отличия метода доступа заключаются в том, что время удержания маркера в сети FDDI не является постоянной величиной, как в сети Token Ring. Это время зависит от загрузки кольца - при небольшой загрузке оно увеличивается, а при больших перегрузках может уменьшаться до нуля. Эти изменения в методе доступа касаются только асинхронного трафика, который не критичен к небольшим задержкам передачи кадров. Для синхронного трафика время удержания маркера по-прежнему остается фиксированной величиной. Механизм приоритетов кадров, аналогичный принятому в технологии Token Ring, в технологии FDDI отсутствует. Разработчики технологии решили, что деление трафика на 8 уровней приоритетов избыточно и достаточно разделить трафик на два класса - асинхронный и синхронный, последний из которых обслуживается всегда, даже при перегрузках кольца.

В остальном пересылка кадров между станциями кольца на уровне MAC полностью соответствует технологии Token Ring. Станции FDDI применяют алгоритм раннего освобождения маркера, как и сети Token Ring со скоростью 16 Мбит/с.

Скорость передачи данных для сетей FDDI составляет 100 Мбит/с. Максимальное число узлов составляет 500. При использовании в качестве физической среды передачи данных многомодового оптоволоконного кабеля расстояние между узлами сети может доходить до 2 км, а при использовании одномодового кабеля – до 40 км. В случае кабеля на основе витых пар пятой категории это расстояние не превышает 100 м. максимальный диаметр двойного кольца не должен превышать 100 км.

Адреса уровня MAC имеют стандартный для технологий IEEE 802 формат. Формат кадра FDDI близок к формату кадра Token Ring, основные отличия заключаются в отсутствии полей приоритетов.

Особенности метода доступа FDDI.

Для передачи синхронных кадров станция всегда имеет право захватить маркер при его поступлении. При этом время удержания маркера имеет заранее заданную фиксированную величину.

Если же станции кольца FDDI нужно передать асинхронный кадр (тип кадра определяется протоколами верхних уровней), то для выяснения возможности захвата маркера при его очередном поступлении станция должна измерить интервал времени, который прошел с момента предыдущего прихода маркера. Этот интервал называется временем оборота маркера. Если в технологии Token Ring максимально допустимое время оборота маркера является фиксированной величиной (2,6 сиз расчета 260 станций в кольце), то в технологии FDDI станции договариваются о его величине во время инициализации кольца. Каждая станция может предложить свое значение, в результате для кольца устанавливается минимальное из предложенных станциями времен. Это позволяет учитывать потребности приложений, работающих на станциях. Обычно синхронным приложениям (приложениям реального времени) нужно чаще передавать данные в сеть небольшими порциями, а асинхронным приложениям лучше получать доступ к сети реже, но большими порциями. Предпочтение отдается станциям, передающим синхронный трафик.

Для обеспечения отказоустойчивости в стандарте FDDI предусмотрено создание двух оптоволоконных колец - первичного и вторичного. В стандарте FDDI допускаются два вида подсоединения станций к сети. Одновременное подключение к первичному и вторичному кольцам называется двойным подключением - Dual Attachment, DA. Подключение только к первичному кольцу называется одиночным подключением - Single Attachment, SA.

В стандарте FDDI предусмотрено наличие в сети конечных узлов – станций и также концентраторов. Для станций и концентраторов допустим любой вид подключения к сети - как одиночный, так и двойной. Обычно концентраторы имеют двойное подключение, а станции -

одинарное, как это показано на рисунке 2, хотя это и не обязательно. Чтобы устройства легче было правильно присоединять к сети, их разъемы маркируются. Разъемы типа A и B должны быть у устройств с двойным подключением, разъем M (Master) имеется у концентратора для одиночного подключения станции, у которой ответный разъем должен иметь тип S (Slave).

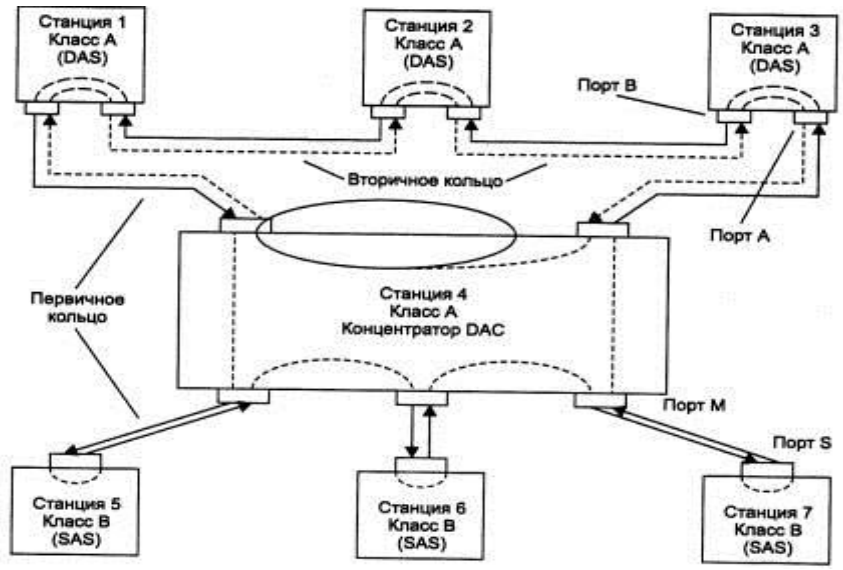


Рисунок 2 – Подключение узлов к кольцам FDDI. SAS (Single Attachment Station), DAS (Dual Attachment Station), SAC (Single Attachment Concentrator) и DAC (Dual Attachment Concentrator).

Стандарт **FDDI** (*Fiber Distributed Data Interface*□ волоконно-оптический интерфейс передачи данных), разработанный в середине 80-х годов комитетом X3T9.5 ANSI, определяет кольцевую сеть с маркерным доступом и скоростью передачи до 100 Мбит/с на основе волоконно-оптического кабеля, способною охватить очень большую площадь (до 100 км).
Стандарт FDDI во многом основывается на технологии Token Ring (стандарт IEEE 802.5) и обеспечивает совместимость с ней, т.к. у обеих технологий одинаковые форматы кадров. Однако у этих технологий имеются существенные различия.

Стек FDDI определяет физический уровень и подуровень доступа к среде передачи

(MAC). Физический уровень разбит на протокол физического уровня (Physical Layer Protocol, PHY), который отвечает за работу схем кодирования данных, и на подуровень физического уровня, зависящий от среды передачи (Physical Medium Dependent, PMD), на котором реализованы спецификации передачи. Особенностью стека FDDI является наличие уровня управления станциями (Station Management, SMT). Он отвечает за удаление и подключение рабочих станций, обнаружение и устранение неисправностей, сбор статистической информации о работе сети.

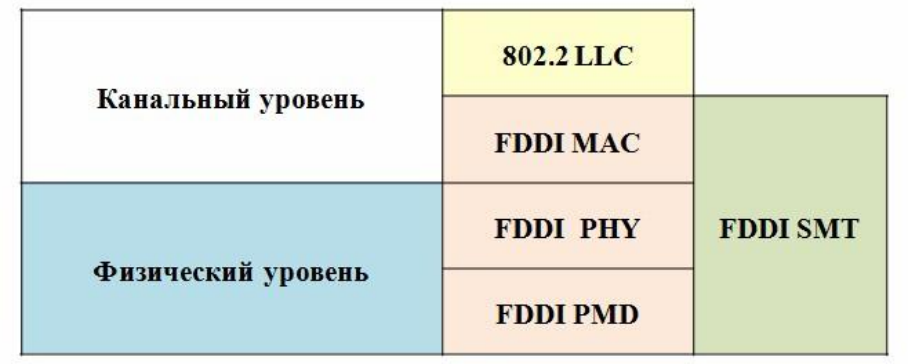


Рисунок 3.Стек FDDI

Сети FDDI характеризуются встроенной избыточностью, что обеспечивает их высокую отказоустойчивость. Сеть FDDI строится на основе двух колец, которые образуют основной и резервный пути передачи данных между узлами сети. Данные в кольцах циркулируют в разных направлениях. Одно кольцо считается основным (первичным). По нему данные передаются при нормальной работе. Второе кольцо (вторичное) вспомогательное, по нему данные передаются в случае обрыва в первом кольце. В случае какого-либо вида отказа, когда часть первого кольца не может передавать данные (например, обрыв кабеля или отказ узла), сеть выполняет «свертывание» колец объединяет первое кольцо со вторым, образуя единое кольцо.

Основными компонентами сети FDDI являются станции и концентраторы. Для подключения станций и концентраторов к сети может быть использован один из двух способов:

- **Одиночное подключение** (Single Attachment, SA) подключение только к первичному кольцу. Станция и концентратор, подключенные данным способом, называются соответственно станцией одиночного подключения (Single Attachment Station, SAS) и концентратором одиночного подключения (Single Attachment Concentrator, SAC).
- **Двойное подключение** (Dual Attachment, DA) одновременное подключение к первичному и вторичному кольцам. Станция и концентратор, подключенные таким способом, называются соответственно станцией двойного подключения (Dual Attachment Station, DAS) и концентратором двойного подключения (Dual Attachment Concentrator, DAC).

В качестве среды передачи в сетях FDDI используется одномодовый и многомодовый волоконно-оптический кабель. Максимальное количество станций в кольце – 500. Максимальное расстояние между узлами может составлять 2 км при использовании многомодового кабеля и 20 км – при использовании одномодового. Максимальная протяженность сети – 100 км.

К преимуществам технологии FDDI можно отнести высокую отказоустойчивость. К недостаткам – двойной расход кабеля.

В настоящее время эта технология считается устаревшей.

Контрольные вопросы

1) Что означает аббревиатура FDDI?
2) Объясните метод доступа в сетях FDDI.
3) Как повышается отказоустойчивость в сетях FDDI?
4) Сравните технологии FDDI и Token Ring (в таблице).

1.18 Лекция №19. Архитектура ATM.

1.18.1 Вопросы лекции:

1. Изучить технологию ATMI;
2. Ознакомиться с особенностями ATM.

1.18.2 Краткое содержание вопросов:

Корпоративные сетевые стандарты позволяют обеспечить эффективное взаимодействие всех станций сети за счет использования одинаковых версий программ и однотипной конфигурации. Однако, значительные сложности возникают при унификации технологии доступа рабочих станций к WAN-сервису, поскольку в этом случае происходит преобразование данных из формата token ring или Ethernet в форматы типа X.25 или T1/E1. ATM обеспечивает связь между станциями одной сети или передачу данных через WAN-сети без изменения формата ячеек - технология ATM является универсальным решением для ЛВС и телекоммуникаций.

Нет сомнений в том, что скоростные технологии ЛВС являются основой современных сетей. ATM, FDDI и Fast Ethernet являются основными вариантами дляорганизация сетей с учетом перспективы. Очевидно, что приложениям multimedia, системам обработки изображений, CAD/CAM, Internet и др. требуется широкополосный доступ в сеть с рабочих станций. Все современные технологии обеспечивают высокую скорость доступа для рабочих станций, но только ATM обеспечивает эффективную связь между локальными и WAN-сетями.

ATM - история и базовые принципы

Технология ATM сначала рассматривалась исключительно как способ снижения телекоммуникационных расходов, возможность использования в ЛВС просто не принималась во внимание. Большинство широкополосных приложений отличается взрывным характером трафика. Высокопроизводительные приложения типа ЛВС клиент-сервер требуют высокой скорости передачи в активном состоянии и практически не используют сеть в остальное время. При этом система находится в активном состоянии (обмен данными) достаточно малое время. Даже в тех случаях, когда пользователям реально не нужна обеспечиваемая сетью полоса, традиционные технологии ЛВС все равно ее выделяют. Следовательно, пользователям приходится платить за излишнюю полосу. Перевод распределенных сетей на технологию ATM позволяет избавиться от таких ненужных расходов.

Комитеты по стандартизации рассматривали решения для обеспечения недорогих широкополосных систем связи в начале 80-х годов. Важно то, что целью этого рассмотрения было применение принципов коммутации пакетов или статистического мультиплексирования, которые так эффективно обеспечивают передачу данных, к системам передачи других типов трафика. Вместо выделения специальных сетевых ресурсов для каждого соединения сети с коммутацией пакетов выделяют ресурсы по запросам (сеансовые соединения). Поскольку для каждого соединения ресурсы выделяются только на время их реального использования, не возникает больших проблем из-за спада трафика.

Проблема, однако, состоит в том, что статистическое мультиплексирование не обеспечивает гарантированного выделения полосы для приложений. Если множество пользователей одновременно захотят использовать сетевые ресурсы, кому-то может просто не хватить полосы. Таким образом, статистическое мультиплексирование, весьма эффективное для передачи данных (где не требуется обеспечивать гарантированную незначительную задержку), оказывается малопригодным для систем реального времени (передача голоса или видео). Технология ATM позволяет решить эту проблему.

Проблема задержек при статистическом мультиплексировании связана в частности с большим и непостоянным размером передаваемых по сети пакетов информации. Возможна задержка небольших пакетов важной информации из-за передачи больших пакетов малозначимых данных. Если небольшой задержанный пакет оказывается частью слова из телефонного разговора или multimedia-презентации, эффект задержки может оказаться весьма существенным и заметным для пользователя. По этой причине многие специалисты считают, что статистическое мультиплексирование кадров данных дает слишком сильную дрожь из-за вариации задержки (delay jitter) и не позволяет предсказать время доставки. С этой точки зрения технология коммутации пакетов является совершенно неприемлемой для передачи трафика типа голоса или видео.

ATM решает эту проблему за счет деления информации любого типа на небольшие ячейки фиксированной длины. Ячейка ATM имеет размер 53 байта, пять из которых составляют заголовок, оставшиеся 48 - собственно информацию. В сетях ATM данные должны вводиться в форме ячеек или преобразовываться в ячейки с помощью функций адаптации. Сети ATM состоят из коммутаторов, соединенных транковыми каналами ATM. Краевые коммутаторы, к которым подключаются пользовательские устройства, обеспечивают функции адаптации, если ATM не используется вплоть до пользовательских станций. Другие коммутаторы, расположенные в центре сети, обеспечивают перенос ячеек, разделение транков и распределение потоков данных. В точке приема функции адаптации восстанавливают из ячеек исходный поток данных и передают его устройству-получателю, как показано на рисунке 4.1.

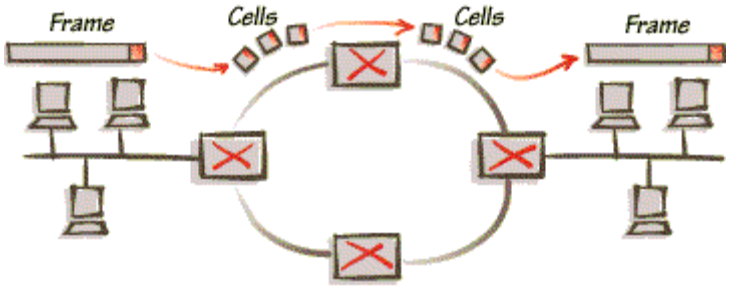


Рисунок 1 Адаптация ATM

Передача данных в коротких ячейках позволяет ATM эффективно управлять потоками различной информации и обеспечивает возможность приоритизации трафика.

Пусть два устройства передают в сеть ATM данные, срочность доставки которых различается (например, голос и трафик ЛВС). Сначала каждый из отправителей делит передаваемые данные на ячейки. Даже после того, как данные от одного из отправителей будут приниматься в сеть, они могут чередоваться с более срочной информацией. Чередование может осуществляться на уровне целых ячеек и малые размеры последних обеспечивают в любом случае непродолжительную задержку.такое решение позволяет передавать срочный трафик практически без задержек, приостанавливая на это время передачу некритичной к задержкам информации. В результате ATM может обеспечивать эффективную передачу всех типов трафика.

Даже при чередовании и приоритизации ячеек в сетях ATM могут наступать ситуации насыщения пропускной способности. Для сохранения минимальной задержки даже в таких случаях ATM может отбрасывать отдельные ячейки при насыщении. Реализация стратегии отбрасывания ячеек зависит от производителя оборудования ATM, но в общем случае обычно отбрасываются ячейки с низким приоритетом (например, данные) для которых достаточно просто повторить передачу без потери информации. Коммутаторы ATM с расширенными функциями могут при отбрасывании ячеек, являющихся частью большого пакета, обеспечить отбрасывание и оставшихся ячеек из этого пакета - такой подход позволяет дополнительно снизить уровень насыщения и избавиться от излишнего объема повторной передачи. Правила отбрасывания ячеек, задержки данных и т.п. определяются набором параметров, называемым качеством обслуживания (Quality of Service) или QoS. Разным приложениям требуется различный уровень QoS и ATM может обеспечить этот уровень.

Поскольку приходящие из разных источников ячейки могут содержать голос, данные и видео, требуется обеспечить независимый контроль для передачи всех типов трафика. Для решения этой задачи используется концепция виртуальных устройств. Виртуальным устройством называется связанный набор сетевых ресурсов, который выглядит как реальное соединение между пользователями, но на самом деле является частью разделяемого множеством пользователей оборудования. Для того, чтобы сделать связь пользователей с сетями ATM как можно более эффективной, виртуальные устройства включают пользовательское оборудование, средства доступа в сеть и собственно сеть ATM.

В заголовке ATM виртуальный канал обозначается комбинацией двух полей - VPI (идентификатор виртуального пути) и VCI (идентификатор виртуального канала. Виртуальный путь применяется в тех случаях, когда 2 пользователя ATM имеют свои собственные коммутаторы на каждом конце пути и могут, следовательно, организовывать и поддерживать свои виртуальные соединения. Виртуальный путь напоминает канал, содержащий множество кабелей, по каждому из которых может быть организовано виртуальное соединение.

Поскольку виртуальные устройства подобны реальным, они также могут быть "выделенными" или "коммутируемыми". В сетях ATM "выделенные" соединения называются постоянными виртуальными устройствами (PVC), создаваемыми по соглашению между пользователем и оператором (подобно выделенной телефонной линии). Коммутируемые соединения ATM используют коммутируемые виртуальные устройства (SVC), которые устанавливаются путем передачи специальных сигналов между пользователем и сетью.

Протокол, используемый ATM для управления виртуальными устройствами подобен протоколу ISDN. Вариант для ISDN описан в стандарте Q.931, ATM - в Q.2931.

Виртуальные устройства ATM поддерживаются за счет мультиплексирования трафика, что существенно снижает расходы на организацию и поддержку магистральных сетей.если в одном из виртуальных устройств уровень трафика невысок, другое устройство может использовать часть свободных возможностей. За счет этого обеспечивается высокий уровень эффективности использования пропускной способности ATM и снижаются цены. Небольшие ячейки фиксированной длины позволяют сетям ATM обеспечить быструю передачу критичного к задержкам трафика (например, голосового). Кроме того, фиксированный размер ячеек обеспечивает практически постоянную задержку, позволяя эмулировать устройства с фиксированной скоростью передачи типа T1E1. Фактически, ATM может эмулировать все существующие сегодня типы сервиса и обеспечивать новые услуги. ATM обеспечивает несколько классов обслуживания, каждый из которых имеет свою спецификацию QoS.

Класс QoS	**Класс обслуживания**	**Описание**
1	A	производительность частных цифровых линий (эмуляция устройств или CBR)
2	B	пакетные аудио/видео-конференции и multimedia (rt-VBR)
3	C	ориентированные на соединения протоколы типа frame relay (nrt-VBR)
4	D	протоколы без организации соединений типа IP, эмуляция ЛВС (ABR)
5	Unspecified	наилучшие возможности в соответствии с определением оператора (UBR)

Большая часть трафика, передаваемого через сети ATM использует класс обслуживания C, X или Y. Класс C определяет параметры QoS (качество обслуживания) для задержки и вероятности отбрасывания, но требует от пользователя аккуратного управления трафиком во избежание перенасыщения сети.трафик класса X дает пользователю большую свободу, но может не обеспечить стабильной производительности. Класс Y, называемый также "Available Bit Rate" (ABR или доступная скорость) позволяет пользователю и сети установить совместно скорость на основе оценки потребностей пользователя и возможностей сети.

ATM как технология ЛВС

Технология ATM изначально создавалась как часть сервиса "Broadband ISDN" под эгидой CCITT (сейчас ITU). Однако возможности ATM можно эффективно использовать и в локальных сетях.

Современные крупные сети используются для передачи самых разных типов данных, включая изображения, звук, CAD/CAM и т.п. Несмотря на то, что большинство компьютерных приложений используется уже достаточно давно, возможности современных настольных компьютеров позволяют по новому подойти к организации работы. Однако, рост возможностей настольных компьютеров существенно опережает расширение сетевых возможностей (в частности, пропускной способности сетей).

Возьмем для примера издательские системы, где с одним набором данных может одновременно работать множество людей. Представьте себе процесс подготовки газетной полосы для публикации.редакторы работают с одной частью полосы, корректоры просматривают текст, дизайнеры размещают материал на полосе - и все это происходит в одно время. Не будем забывать и о том, что высокое качество печати требует использования графических фалов размером в сотни мегабайт. Традиционные сети обеспечивают разделение доступа к таким файлам, однако из-за ограниченной пропускной способности доступ к

расположенному на другом компьютере файлу размером в несколько сот мегабайт будет отнюдь не быстрым. ATM 25 позволяет пользователям организовать каналы доступа с полосой 25 Мбит/с для работы с серверами. Такое решение избавляет от задержек и позволяет готовить публикации существенно быстрее.

Преимущества ATM не ограничиваются вертикальным рынком. Сегодня организации могут связать через магистрали ATM свои корпоративные серверы. Можно ожидать и достаточно широкого использования ATM в настольных компьютерах при работе пользователей с большими объемами данных или использовании критичных к задержкам приложений.

Экономический фактор играет далеко не последнюю роль в расширении использования технологий ATM. Сегодня большинство людей использует в своей работе и телефон и компьютер. В течение нескольких лет существенно расширится обмен данными multimedia (клипами), использование видеоконференций и т.п. технология ISDN позволяет решить такие задачи. Однако, это потребует установки оборудования ISDN в каждый компьютер. Телефонные и сетевые кабельные системы не могут полностью совпадать, что дополнительно увеличивает сложность такого решения. Использование решения на базе ISDN с необходимостью приведет к возникновению параллельных кабельных систем для ЛВС и телефонии и подключению каждого компьютера к обеим системам. Нужно учесть еще и телевизионные кабели, которые также требуется проложить по причине расширения использования настольных видео-приложений. Такая кабельная система будет весьма сложна, а ее установка и поддержка потребуют высоких расходов. Переход на использование технологии ATM в локальных сетях позволяет обойтись одной кабельной системой и одним адаптером в компьютере, что не может не привести к значительному снижению расходов.

ATM позволяет не только организовать ЛВС, но может обеспечить передачу голосового и видео-трафика. Такое решение позволяет использовать настольные системы видеоконференций и приложения multimedia.

Фактически, использование ATM обеспечивает сразу множество преимуществ. Во-первых, высокая скорость доступа за приемлемую цену, во-вторых, возможность организации компактных магистралей на базе ATM (collapsed backbone). Наконец, эта архитектура обеспечивает сквозное повышение эффективности использования сетевых ресурсов.

Пользователи, которые думают об использовании ATM в будущем, должны использовать совместимые с ATM устройства уже сегодня - в противном случае переход может оказаться слишком дорогим и трудоемким. Мы рассмотрим этот вопрос более подробно в следующем разделе.

ATM как современная инфраструктура

Если виртуальные устройства напоминают реальные, ATM можно легко приспособить для текущих приложений, просто заменив выделенные или коммутируемые линии виртуальными устройствами ATM. Фактически, этот способ вместе с переходом на ATM в сетевых магистралях, является наиболее очевидным первым шагом.

На рисунках 4.2, 4.3 и 4.4 показан типичный пользовательский сайт с устройствами, порождающими разнотипный трафик (голос, видео, данные). Эти три типа трафика могут передаваться с использованием сервиса ATM тремя показанными на рисунках способами.

1. Голос, данные и видео преобразуются в ячейки ATM в сети оператора с использованием функций адаптации ATM. Оператор будет реализовать все функции доступа и передачи, а для каждого устройства потребуется отдельная линия доступа в сеть ATM.

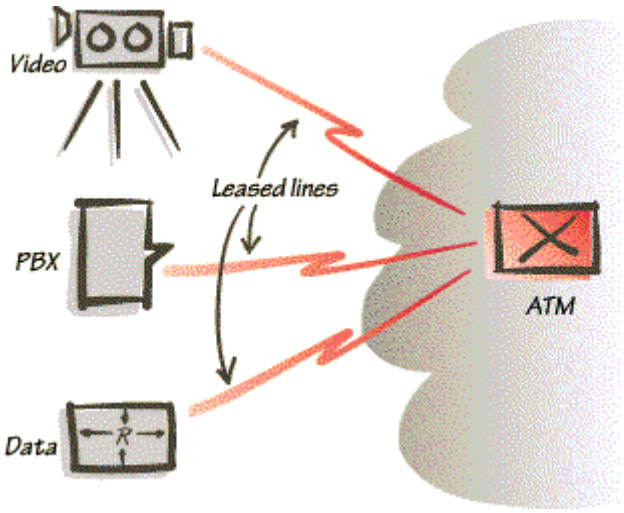


Рисунок 2 Преобразование в ATM осуществляется оператором

2. ЛВС, голосовые и видео-устройства подключаются к локальному коммутатору ATM для преобразования трафика в ячейки. Для доступа в сеть оператора используется одна линия, передающая все потоки трафика одновременно (как виртуальные устройства). Сеть оператора обеспечивает маршрутизацию трафика. Такое решение более экономично и может использоваться для организации "частных сетей ATM" для пользователей, которые имеют доступ к ATM-сервису или хотят создать свою распределенную сеть на базе ATM. Отметим, что находящийся в сети пользователя коммутатор ATM может принадлежать оператору и находиться у него на обслуживании.

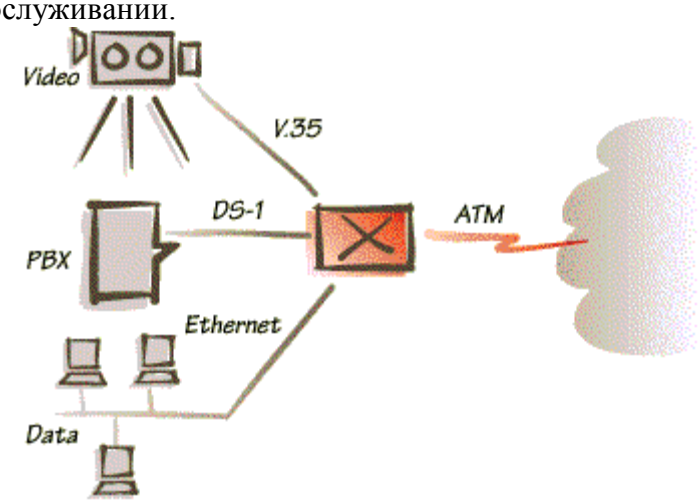


Рисунок 3 Преобразование в ATM осуществляется у пользователя

3. Устройства оборудуются собственными интерфейсами ATM. Одно устройство доступа позволяет объединить весь пользовательский трафик в одном транке, связанном с сетью оператора. В этом случае на стороне пользователя устанавливается принадлежащее ему оборудование ATM, которое можно использовать для организации магистралей ЛВС или подключения настольных станций.

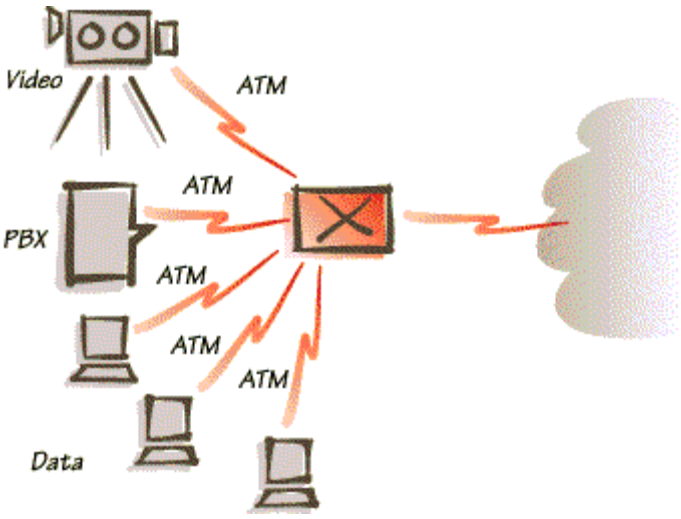


Рисунок 4 Сеть на базе ATM

Скорое появление интерфейсов ATM в телефонном и видео-оборудовании не представляется вероятным, поэтому реализация третьего варианта соединения с сетью не сможет в ближайшие годы стать доминирующей. Фактически, скорость распространения каждого из приведенных вариантов будет определяться темпами снижения цен на оборудование и услуги операторов сетей ATM. Отсутствие эффективного управления этими процессами порождает определенный хаос и не позволяет надежно предсказать перспективы того или иного сервиса ATM.

Стандарт, определяющий интерфейс между операторами и пользователями ATM называется Public User Network Interface или Public UNI. Этот интерфейс определяется для различных значений скорости. Первые услуги ATM предлагались в основном со скоростью T3 (45 Мбит/с). Сейчас многие операторы предлагают скорость 155 Мбит/с и выше, но такая полоса обычно не требуется пользователям, да и стоимость подобных услуг весьма высока. Для большинства пользователей, планирующих организовать доступ к ATM или создать частную сеть ATM основной проблемой является стоимость оборудования.

Форум ATM - организация производителей оборудования ATM и пользователей работает в направлении развития стандартов и обеспечения интероперабельности оборудования. В конечном итоге это не может не привести к снижению цен. Кроме обеспечения интероперабельности ATM ведется большая работа по реализации ATM на скоростях меньше T3. Здесь возможно несколько вариантов:

1. Полнофункциональные решения ATM при скорости T1. Один стандарт для ATM T1 уже утвержден, но некоторые производители и пользователи считают, что связанные с реализацией этого стандарта накладные расходы слишком велики - канал T1 с полосой 1.544 Мбит/с может обеспечить полезную полосу только около 1.1 Мбит/с.
2. Так называемый dixie-стандарт (от акронима DXI - Data eXchange Interface). DXI был разработан как способ использования ATM в кадровом режиме с маршрутизаторами и другими устройствами передачи данных и специальными устройствами DSU, обеспечивающими преобразование кадров в реальные ячейки ATM. DXI работает через стандартные интерфейсы типа V.35 и HSSI.
3. Интерфейс пользователь - сеть Frame Relay или F-UNI (произностися как FOONY), являющийся стандартом использования frame relay для доставки "кадров данных ATM" в сеть, которая будет конвертировать их в ячейки непосредственно на границе сети.
4. Инверсное мультиплексирование ATM или AIM - стандарт для инверсного мультиплексирования множества линий T1 в один транк с полосой между T1 и T3. Такая полоса обеспечивает поддержку ATM для приложений, где скоростные запросы незначительно превышают возможности T1.

Проверка этих вариантов показывает, что они в основном подходят для систем обмена данными. Причиной этого является эффективная поддержка технологией ATM взрывного

трафика современных систем передачи данных (ЛВС). Как было отмечено выше ATM может просто использоваться взамен выделенных линий в таких сетях, обеспечивая коммутацию ЛВС, поддерживаемую ATM UNI.

Замена выделенных линий системами ATM позволяет более эффективно организовать сети. Отметим, что виртуальные устройства ATM используются для организации многосвязных систем, позволяющих обеспечить доставку трафика непосредственно адресату. Сегодня желание пользователей применять многосвязные системы на базе ATM для связи своих сетей в значительной мере определяется предлагаемыми операторами ценами на услуги. Если оператор берет деньги за каждое виртуальное устройство ATM UNI, а не за общий трафик, стоимость организации многосвязной сети может оказаться слишком велика. Конечно, в кампусной магистрали ATM стоимость полосы в многосвязной системе будет несравненно ниже. Поддержка многосвязности требует лишь прокладки дополнительных физических соединений (кабелей) и установки более скоростных транковых портов в коммутаторы. Эти дополнительные расходы достаточно малы по сравнению с общей стоимостью сети.

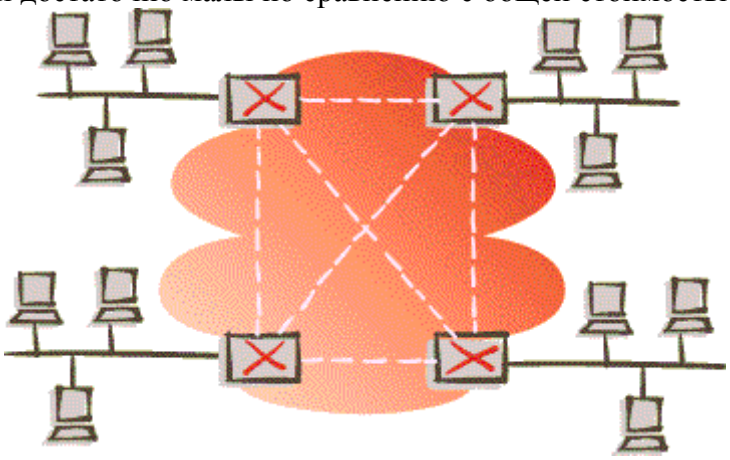


Рисунок 5 Многосвязная сеть

В многосвязной (каждый с каждым) сети ATM существует меньше транзитных точек, снижающих производительность и вносящих дополнительные задержки и насыщение. Такое решение обеспечивает существенное повышение стабильности работы приложений. Более того, каждый коммутатор является соседом для всех остальных коммутаторов и связан с ними напрямую. Это упрощает задачу динамического определения маршрута для протоколов маршрутизации типа RIP, используемого TCP/IP или NetWare, OSPF или IS-IS. Эти протоколы часто генерируют значительный трафик и могут существенно замедлить сеть при обмене конфигурационными данными (интервал сближения или конвергенции).

Если существует способ передачи "телефонного номера" ATM точке публичной сети ATM, которая достаточно близка к пользователям традиционной ЛВС, насколько можно приблизиться к пользователям? Ближайшей, готовой к использованию ATM станцией, сегодня является коммутатор. Это может быть магистральный коммутатор пользователя, коммутатор рабочей группы или даже настольный компьютер с адаптером ATM. В этом случае ATM используется как универсальная архитектура для коммуникаций, обеспечивающая связь между настольными системами вместе с традиционными технологиями ЛВС, а в некоторых случаях - взамен их. Это наиболее интересная, но и наиболее спорная часть применений ATM.

Сквозная ATM-парадигма для сетей

ATM на настольных станциях имеет несколько преимуществ. Во-первых, способность ATM гарантировать для приложений качество обслуживания (QoS) обеспечивает сквозную передачу критичного к задержкам трафика типа видео или голоса. Будучи технологией передачи данных, ATM не только может поддерживать "приложения завтрашнего дня", но и эффективно справляется с сегодняшними задачами. Пользователи задаются двумя основными вопросами - как будут формироваться распределенные сети на базе ATM и какие шаги нужно

предпринять, чтобы быть готовым к переходу? Есть три разных варианта включения ATM в архитектуру межсетевого взаимодействия для современных и будущих приложений:

1. Эмуляция традиционных протоколов ЛВС с использованием оборудования ATM. В этом случае существующие приложения будут продолжать работать как раньше, а ATM-добавит к существующим протоколам новые, специально разработанные для приложений multimedia. Отметим, что слово "новые" в данном контексте отнюдь не означает, что эти протоколы еще не существуют (они скорее еще не стали общепринятыми).
2. Подключение сервиса ATM напрямую к интерфейсам прикладных программ, используемых сегодня, в обход традиционных протоколов нижних уровней. Для поддержки этого варианта потребуется разработка новых API.
3. Использование новых API для "новых" приложений и эмуляция традиционных протоколов для существующих приложений.

Поскольку использование ATM обычно начинается с нескольких станций, которым требуются multimedia-приложения, требуется обеспечить эмуляцию традиционных протоколов ЛВС в сетях ATM. Это позволяет обеспечить надежное взаимодействие между новыми станциями на базе ATM и традиционными ЛВС. Для эмуляции ЛВС в системах на базе ATM (ATM LAN emulation) предложены два варианта - ATM Forum LAN Emulation (LANE) и RFC 1577. Говоря здесь об эмуляции, мы имеем в виду оба варианта.

Как LANE, так и RFC 1577 основаны на допущении что пользователи ATM применяют адаптеры, поддерживающие интерфейс ATM UNI. Поскольку этот интерфейс располагается со стороны пользователя, его иногда называют "Private UNI"; существует набор стандартов, определяющих данный интерфейс. Стандарты Private UNI существуют для скоростей 25 Мбит/с (по медному кабелю), 100 Мбит/с (оптический кабель)и 155 Мбит/с (медь и оптика). Оба стандарта эмуляции ЛВС предполагают также, что пользователи подключены к коммутатору ATM. Некоторые ATM-коммутаторы поддерживают также станции других типов (не ATM). Такие коммутаторы обеспечивают взаимодействие между ЛВС Ethernet и token ring и сетями ATM. Коммутаторы также поддерживают порты (для подключения станций и серверов) и транки (для соединения коммутаторов ATM или подключения к магистральным коммутаторам) ATM. Интерфейс между коммутаторами основан на UNI, но включает дополнительно специальные сообщения для маршрутизации и управления состоянием маршрутов. ATM Forum называетэтотинтерфейс Private Network-to-Network Interface или P-NNI.

Эмуляция ЛВС во всех вариантах состоит из двух программных частей - функции клиента используются на конечных системах, подключенных к эмулируемым ЛВС, а функции сервера - реализуются в каждой группе клиентских станций. Группа клиентов и связанный с ней сервер называются эмулируемой ЛВС (Emulated LAN или ELAN).

Протоколы ЛВС являются многоуровневыми и, следовательно, любой стандарт, обеспечивающий взаимодействие традиционных ЛВС и ATM должен обеспечивать поддержку соответствующих уровней. В этом вопросе существующие стандарты эмуляции ЛВС существенно различаются. ATM LANE (стандарт ATM Forum) предназначен для эмуляции протоколов канального (MAC/LLC) уровня. Поскольку этот протокол занимает самый нижний для ЛВС уровень, LANE можно использовать со всеми протоколами ЛВС вышележащих уровней, включая TCP/IP, NetWare SPX/IPX, IBM SNA/LLC2. RFC 1577, с другой стороны, работает на сетевом уровне (уровень 3) и предназначен для протокола TCP/IP.

Оба варианта эмуляции ЛВС похожи по принципам работы, несмотря на различие уровней. При организации ATM ЛВС клиентские системы пытаются вступить в контакт с сервером и зарегистрировать адресную информацию, которая содержит адрес ATM, а также адреса канального и сетевого уровней. Сервер строит каталог адресной информации для последующего использования. По завершении регистрации клиенты и серверы переходят в режим ожидания пользовательского трафика.

Пользовательские программы, работающие на клиентских и серверных системах, функционируют в среде эмуляции ЛВС как в обычных средах традиционных локальных сетей и только коммуникационные драйверы нижних уровней связаны с ATM. Когда программа

генерирует сообщение, это сообщение передается вниз по стеку протоколов программам ATM, прибывая к ним в форме дейтаграммы или сообщения без организации соединения на уровне два (канальном) или уровне 3 (сетевом) в зависимости от способа эмуляции ЛВС. Программы ATM должны обеспечить эмуляцию ЛВС.

Если между отправителем и получателем будет существовать виртуальное устройство, дейтаграммы можно просто помещать в это виртуальное устройство и передавать получателю в исходной форме (дейтаграмма) для обработки на станции получателя программами ATM и приложением. Фактически, каждый клиент ATM поддерживает таблицу адресов канального и сетевого уровня, а же идентификаторов виртуальных устройств ATM (VPI/VCI). Если адрес получателя найден в таблице, дейтаграмма передается соответствующему виртуальному устройству. Проблема возникает когда адрес получателя не найден - в этом случае в игру вступает сервер эмуляции ЛВС.

Клиентская система, не имеющая виртуального устройства ATM, должна организовать его, но дейтаграмма является сообщением ЛВС и не содержит ATM-адреса получателя. Для получения этого адреса клиент посылает сообщение своему серверу, указывая получателя дейтаграммы с помощью адреса сетевого и/или канального уровня и запрашивая соответствующий адрес ATM. Сервер сообщает адрес, после чего клиент организует коммутируемое соединение ATM SVC с адресатом, в которое направляется поток дейтаграмм.

Сервер также обеспечивает поддержку широковещательного и неадресованного (broadcast and unknown) трафика для клиентов, рассылающих широковещательные и групповые (multicast() дейтаграммы. Сервер в таких случаях пересылает принятые дейтаграммы всем зарегистрированным клиентам. Перед организацией SVC клиент может также использовать режим "broadcast and unknown" для рассылки дейтаграмм адресатам, для которых адреса ATM еще не получены.

Устройства традиционных ЛВС должны обмениваться данными со станциями ATM, работающими в эмулируемых ЛВС; коммутаторы обеспечивают функции proxy-клиента от имени станций традиционных ЛВС (не ATM). В этом случае станция ATM, вызывающая станцию ЛВС будет получать от сервера адрес proxy-клиента и организовывать SVC по этому адресу. Proxy-клиент будет в этом случае играть роль моста или маршрутизатора для передачи дейтаграмм нужной станции. На практике такое использование эмуляции является преобладающим, поскольку большинство настольных станций по-прежнему используют Ethernet или token ring.

Это может выглядеть как попытка создания всемирной "плоской" сети, но это не так. RFC 1577 задает ограничение на размер доменов эмуляции ЛВС - не более одной IP-подсети на домен. ATM Forum LANE не содержит такого ограничения, но практический размер домена устанавливается числом генерируемых многоадресных сообщений (с ростом этого числа растет нагрузка на сервер и клиентов). В действительности LANE представляет собой мост, а широковещательный и групповой трафик всегда является ограничивающим фактором для сетей на базе мостов.

Как связать между собой эмулируемые домены ЛВС? Лучшим способом является использование коммутаторов ЛВС. Поскольку коммутатор может одновременно работать с ATM LANE и дейтаграммами традиционных ЛВС, он может обеспечивать связь эмулируемых доменов (как подсетей IP или сегментов ЛВС).

Проблема возникает при использовании маршрутизаторов для соединения устройств ATM, использующих multimedia-приложения. Маршрутизаторы, как устройства, работающие без организации соединений, не могут обеспечивать гарантии качества обслуживания (QoS), предлагаемой коммутаторами ATM. Таким образом, маршрутизатор между двумя станциями ATM существенно ограничивает возможности связи между этими станциями (до уровня станций традиционных ЛВС). Решения на базе коммутаторов позволяют сохранить гибкость и скорость ATM.

Естественные соединения ATM требуют коммутируемого пути между адресатом и отправителем. Если оба устройства подключены к одному коммутатору, проблем не возникает. Также просто организовать связь между устройствами, использующими услуги одного

оператора или коммутаторы одного производителя. При соединении устройств в среде с разнотипным оборудованием может потребоваться использование PNNI для организации мостов между двумя или несколькими коммутаторами ATM и в тех случаях, когда ATM-соединение организуется через распределенную сеть (WAN).

Существует три варианта организации "реальных" соединений ATM через распределенную сеть:

1. Выделенная цифровая линия от оператора (T3, например) служить транком между двумя коммутаторами ATM - эти коммутаторы будут генерировать ячейки, обеспечивать сигнализацию ATM и поддерживать потоки трафика. Фактически, это вариант частной сети ATM.
2. Оператор ATM может обеспечивать виртуальный путь между парой коммутаторов. В этом случае оператор передает ячейки и принимает участие в управлении трафиком ATM, но соединенные между собой устройства управляются виртуальными устройствами как при использовании соединения по выделенной линии.
3. Может использоваться предоставляемое оператором коммутируемое соединение ATM SVC.

В первых двух вариантах ATM-коммутаторы принадлежат пользователю и должны выполнять все операции по преобразованию адресов (логические адреса, известные приложениям, конвертируются в реальные адреса ATM). В последнем варианте может потребоваться преобразование адресов оператором или, по крайней мере, использование архитектуры, поддерживающей соединений частных сетей через публичные. Одна из таких архитектур обеспечивается протоколом NHRP (маршрутизация в следующий интервал), предложенным IETF. Поскольку элементы протокола NHRP включены в базовую архитектуру стандарта ATM Forum MPOA, очевидно, что MPOA будет поддерживать управление адресами в больших сетях ATM, подключенных к системам общего пользования.

В долгосрочной перспективе ATM может полностью заменить технологии ЛВС и системы межсетевого взаимодействия в их современном виде. Сети на базе коммутаторов, в результате, будут значительно более гибкими, нежели связанные между собой ЛВС. Стоимость таких решений также может оказаться меньше. Многие пользователи верят в перспективность ATM и даже неизбежность успеха этой технологии. Однако переход к использованию ATM тормозится высокими ценами на оборудование и сложностью его использования.

Эволюция

Большинство организаций входят в одну из трех категорий с точки зрения перспектив использования ATM:

1. Организации, которые используют приложения сильно выигрывающие в результате перехода на ATM. Примером компаний этого класса являются организации здравоохранения, брокерские фирмы с большими потоками коммерческой информации, компании, занимающиеся производством видеопродукции.
2. Организации, которые могут перейти на ATM в результате агрессивной ценовой политики поставщиков услуг.
3. "Оборонительная стратегия" Организации этого типа знают, что технология ATM обеспечит им целый ряд преимуществ, но пока не планируют использовать данную технологию.

Для любой компании первым правилом эволюции ATM является *предотвращение потери средств, вложенных на этапе оценки технологии ATM*. Это означает, что при покупке сетевого оборудования сегодня нужно принимать во внимание возможность использования этого оборудования в будущей сети на базе ATM. Если от закупаемого сегодня оборудования придется потом отказываться, лучше сразу поискать другое решение.

Это правило наиболее ярко проявляется при выборе сетевых коммутаторов. Приобретаемые сегодня устройства должны обеспечивать возможность использования в системах на базе ATM. Минимальным требованием является возможность использования ATM-транков для связи между коммутаторами. Желательно также иметь в коммутаторе порт

(или гнездо для его установки), позволяющий в будущем подключить настольные станции с интерфейсом ATM. Маршрутизаторы, пока не будет найдено более эффективного решения для ATM, должны использоваться как краевые устройства, обеспечивающие возможность подключения устройств традиционных ЛВС к сетям ATM. По крайней мере, такие устройства должны иметь интерфейс proxy-клиента эмуляции ЛВС.

Организации с "оборонной" стратегией, отмеченные в категории три, могут счесть наличие транкового порта ATM в коммутаторе достаточной для ближайших перспектив использования ATM (использовать не будем, но на всякий случай возьмем).

Компании, планирующие для ATM ключевую роль в своей сети, должны выбирать коммутаторы с портами ATM для подключения настольных станций. ATM обеспечивает широкий диапазон скоростей для подключения настольных станций - от 25 до 155 Мбит/с. ATM25 работает с кабельными системами категории 3 - 5 и может использоваться вместо token ring или 10BaseT для станций с высоким уровнем сетевых запросов.

Снижение цен на оборудование ATM для настольных станций играет важную роль, поскольку сегодня приложений, не способных обойтись без возможностей ATM, еще не так много. Скорей всего, пользователи первых станций ATM будут работать с одним из рассмотренных выше вариантов эмуляции ЛВС и большинство приложений будут скорее использовать эмуляцию, нежели естественные ATM API. Адаптеры ATM и коммутационные технологии должны удовлетворять потребности пользователей в течение 5 -8 лет, а скорость отказа от традиционных технологий ЛВС будет в значительной мере определяться темпами расширения числа видеоприложений.

Понимание того, что большинство пользователей не работает с приложениями, требующими возможностей ATM зачастую служит тормозом внедрения ATM, поскольку никому не хочется тратить деньги га приобретение неиспользуемых возможностей. Использование ATM только на части станций избавит от ненужных расходов на модернизацию сети.

Если вы предполагаете начать использование ATM в настольных станциях в течение ближайшей пары лет, вам нужно выбирать коммутаторы с учетом этой перспективы. Коммутаторы должны иметь порты для подключения станций и магистральны порты 155 и 622 Мбит/сдля соединения коммутаторов. Порты ATM должны поддерживать эмуляцию ЛВС. Важно также обратить внимание на перспективы реализации в коммутаторах поддержки таких протоколов, как RFC 1577 и MPOA. наконец, транковый интерфейс для связи с другими коммутаторами должен поддерживать стандарт PNNI.

Если оператор ATM предлагает свои услуги по разумным ценам или ваша организация планирует организовать собственную магистраль ATM, следует оценить потребности до покупки оборудования ATM. Остается ответить на вопрос "Какой тип ATM-сервиса использовать?"

Публичные или частные системы ATM будут нормально поддерживать подключение устройств frame relay через специальные преобразователи (ATM DSU/CSU). Если ваше соглашение с оператором ATM требует покупки такого оборудования для подключения других источников трафика к ATM, может оказаться более эффективной реализация сервиса frame relay на базе существующих коммутаторов и их связь с ATM через краевые устройства.

Если для подключения связывающих сети устройств (типа маршрутизаторов) к ATM вам потребуется покупать дополнительные устройства, лучше будет купить интерфейс ATM для коммутатора. Этот интерфейс можно будет использовать и после перехода на ATM, тогда как устройства DSU/CSU после такого перехода станут просто ненужными. Существует три варианта подключения ATM к коммутаторам:

1. Естественная форма ATM (ячейки) с прямым подключением цифрового транка ATM (обычно T1 или T3) к маршрутизатору. Этот тип интерфейса может поддерживать все типы сервиса ATM (включая multimedia). Такой вариант целесообразно выбирать при планировании перехода от маршрутизаторов к коммутаторам ATM.
2. DXI-форма ATM - интерфейс на основе кадров, поддерживающий только транспортный сервис ATM, ориентированный на передачу данных. Такой тип подключения

хорош для систем, где не планируется замена маршрутизаторов на коммутаторы ATM. Выбирая этот вариант, следует помнить, что некоторые операторы ATM не поддерживают DXI-сервис и может потребоваться покупка ATM DSU/CSU для преобразования DXI в ячейки ATM.

3. Интерфейс F-UNI, который представляет собой вариант интерфейса frame relay с поддержкой сигнализации ATM. Этот вариант пока распространен недостаточно широко, но может обеспечить просто и недорогой переход для маршрутизаторов, которые уже поддерживают frame relay.

При любом варианте перехода на ATM в первую очередь возникает задача организации магистралей. Организация компактных магистралей (collapsed backbone) без использования технологии ATM в таком случае будет весьма рискованным решением. Магистральные технологии при переходе на ATM приходится менять в первую очередь. Наиболее критичным при переходе на ATM будет первый шаг в сторону от традиционной коммутации ЛВС. В системах коммутации ЛВС без ATM-транков магистрали не используют технологии ATM и, следовательно, модернизация магистралей будет достаточно рискованным шагом. В идеальном случае коммутаторы ЛВС должны поддерживать магистрали ATM и других типов (например, FDDI).

Переход приложений на ATM будет постепенным. На настольных станциях ATM будет поначалу использоваться для эмуляции ЛВС и работы с набором традиционных приложений ЛВС. По мере расширения инфраструктуры ATM станет возможным связать большие группы пользователей в "чистые" сети ATM. Это позволит использовать специальные приложения, рассчитанные на качество обслуживания ATM (видео, multimedia и т.п.) или упростить работу с традиционными потоками данных за счет более высокой производительности ATM.

ATM, по мере реализации, будет делать сеть компании более гармоничной - сначала на уровне магистралей, а потом и для настольных систем. Полный переход на ATM наверняка будет определяться темпами снижения цен на порты для подключения настольных станций и адаптеры, а также реализацией поддержки возможностей в прикладных программах. Использование единой технологии для организации магистралей, подключения настольных станций и распределенных сетей может обеспечить, в конечном итоге, существенную экономию.

В долгосрочной перспективе ATM должна стать единой архитектурой внутрикорпоративных и межкорпоративных коммуникаций. Коммутируемые виртуальные устройства, используемые настольными системами могут быть расширены за счет поддержки соединений SVC операторами публичных сетей, делая ATM универсальной технологией multimedia-сетей. Протоколы типа NHRP являются средством обеспечения универсальной связи, но в конечном итоге набор протоколов ATM для multimedia будет, по-видимому, основан на службах каталогов.

Степень воздействия универсальных multimedia-коммуникаций на бизнес достаточно трудно прогнозировать с учетом отсутствия альтернативных вариантов. Несомненно, ATM будет играть значительную роль в коммерции, здравоохранении, обучении за счет систем распространения информации. Системы ATM основаны на экономичной технологии мультиплексирования, позволяющей преодолеть барьеры, связанные с взрывным характером трафика во многих приложениях.

С учетом всех этих влияний технология ATM остается привлекательной реализацией и очевидно, что множество пользователей будут готовы перейти на ATM в ближайшем будущем. Это означает, что и ваша организация может быстро начать работу с ATM и расширять использование этой технологии для повышения эффективности работы.

2. Методические указания по выполнению лабораторных работ

2.1 Лабораторная работа №1 (2 часа)
Тема: «Сетевая модель OSI»
Цель работы: изучить правила адресации сетевого уровня, научиться распределять адреса между участниками сети передачи данных и организовывать маршрутизацию между сегментами сети. Изучить правила адресации сетевого уровня, научиться распределять адреса между участниками сети передачи данных и организовывать маршрутизацию между сегментами сети.
Задачи работы:
1. изучить сетевую модель OSI;
2. изучить правила адресации сетевого уровня на примере протокола IP;
3. изучить маршрутизацию IP.
Перечень приборов, материалов, используемых в лабораторной работе:
1. персональный компьютер, включенный в сеть IP, Microsoft Windows.
Описание (ход) работы:

Модель OSI описывает только системные средства взаимодействия, реализуемые операционной системой, системными утилитами, системными аппаратными средствами. Модель не включает средства взаимодействия приложений конечных пользователей.

Эта модель описывает функции семи иерархических уровней и интерфейсы взаимодействия между уровнями. Каждый уровень определяется сервисом, который он предоставляет вышестоящему уровню, и протоколом - набором правил и форматов данных для взаимодействия между собой объектов одного уровня, работающих на разных компьютерах.

Каждый уровень поддерживает интерфейсы с выше- и нижележащими уровнями. Ниже перечислены (в направлении сверху вниз) уровни модели OSI и указаны их общие функции.

Уровень приложения (Application) - интерфейс с прикладными процессами.

Уровень представления (Presentation) - согласование представления (форматов, кодировок) данных прикладных процессов.

Сеансовый уровень (Session) - установление, поддержка и закрытие логического сеанса связи между удаленными процессами.

Транспортный уровень (Transport) - обеспечение безошибочного сквозного обмена потоками данных между процессами во время сеанса.

Сетевой уровень (Network) - фрагментация и сборка передаваемых транспортным уровнем данных, маршрутизация и продвижение их по сети от компьютера-отправителя к компьютеру-получателю.

Канальный уровень (Data Link) - управление каналом передачи данных, управление доступом к среде передачи, передача данных по каналу, обнаружение ошибок в канале и их коррекция.

Физический уровень (Physical) - физический интерфейс с каналом передачи данных, представление данных в виде физических сигналов и их кодирование.

Принципы выделения этих уровней таковы: каждый уровень отражает надлежащий уровень абстракции и имеет строго определенную функцию. Эта функция выбиралась, прежде всего, так, чтобы можно было определить международный стандарт. Границы уровней выбирались так, чтобы минимизировать поток информации через интерфейсы.

Два самых низших уровня - физический и канальный - реализуются аппаратными и программными средствами, остальные пять более высоких уровней реализуются, как правило, программными средствами (рисунок 1).

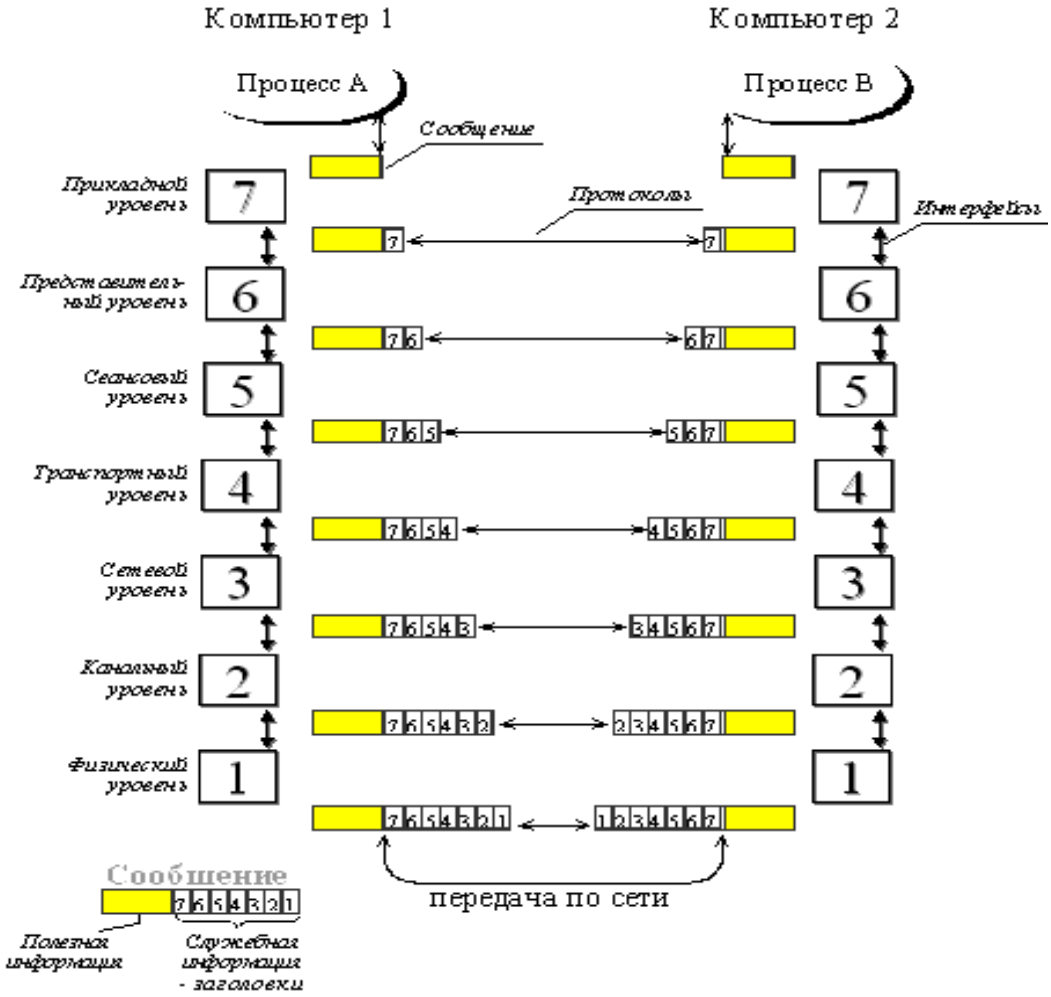


Рисунок 1 - Модель взаимодействия открытых систем ISO/OSI

При продвижении пакета данных по уровням сверху вниз каждый новый уровень добавляет к пакету свою служебную информацию в виде заголовка и, возможно, трейлера (информации, помещаемой в конец сообщения). Эта операция называется инкапсуляцией данных верхнего уровня в пакете нижнего уровня. Служебная информация предназначается для объекта того же уровня на удаленном компьютере, ее формат и интерпретация определяются протоколом данного уровня. Наконец, сообщение достигает нижнего, физического уровня, который, собственно, и передает его по линиям связи машине-адресату. К этому моменту сообщение "обрастает" заголовками всех уровней.

Когда сообщение по сети поступает на другую машину, оно последовательно перемещается вверх с уровня на уровень. Каждый уровень анализирует, обрабатывает и удаляет заголовок своего уровня, выполняет соответствующие данному уровню функции и передает сообщение вышележащему уровню. Тот в свою очередь рассматривает эти данные как пакет со своей служебной информацией и данными для верхнего уровня, и процедура повторяется, пока пользовательские данные, очищенные от всей служебной информации, не достигнут прикладного процесса.

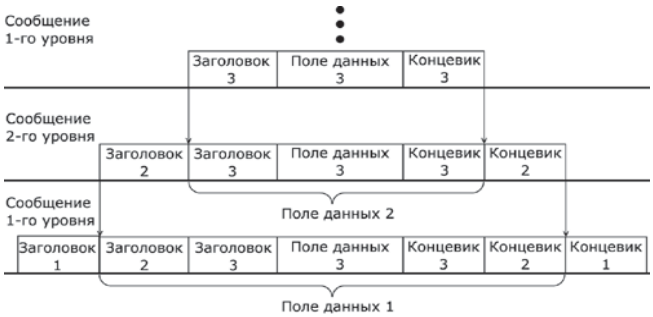


Рисунок 2 – Вложенность сообщений различных уровней

Кроме термина "сообщение" (message) существуют и другие названия, используемые сетевыми специалистами для обозначения единицы обмена данными. В стандартах ISO для протоколов любого уровня используется такой термин как "протокольный блок данных" - Protocol Data Unit (PDU). Кроме этого, часто используются названия кадр (frame), пакет (packet), дейтаграмма (datagram).

Теперь рассмотрим каждый уровень этой модели. Отметим что это модель, а не архитектура сети. Она не определяет протоколов и сервис каждого уровня. Она лишь говорит, что он должен делать.

Физический уровень. Этот уровень имеет дело с передачей битов по физическим каналам, таким, например, как коаксиальный кабель, витая пара или оптоволоконный кабель. К этому уровню имеют отношение характеристики физических сред передачи данных, такие как полоса пропускания, помехозащищенность, волновое сопротивление и другие. На этом же уровне определяются характеристики электрических сигналов, такие как требования к фронтам импульсов, уровням напряжения или тока передаваемого сигнала, тип кодирования, скорость передачи сигналов. Кроме этого, здесь стандартизуются типы разъемов и назначение каждого контакта.

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

В обобщенном виде функции физического уровня заключаются в следующем:

- передача битов по физическим каналам;
- формирование электрических сигналов;
- кодирование информации;
- синхронизация;
- модуляция.

Этот уровень реализуется аппаратно. Примером протокола физического уровня может служить спецификация 10Base-T технологии Ethernet.

Канальный уровень. На физическом уровне просто пересылаются биты. При этом не учитывается, что в некоторых сетях, в которых линии связи используются (разделяются) попеременно несколькими парами взаимодействующих компьютеров, физическая среда передачи может быть занята. Поэтому одной из задач канального уровня является проверка доступности среды передачи. Другой задачей канального уровня является реализация механизмов обнаружения и коррекции ошибок. Для этого на канальном уровне биты группируются в наборы, называемые кадрами (frames). Канальный уровень обеспечивает корректность передачи каждого кадра, помещая специальную последовательность бит в начало и конец каждого кадра, чтобы отметить его, а также вычисляет контрольную сумму, суммируя все байты кадра определенным способом и добавляя контрольную сумму к кадру. Когда кадр приходит, получатель снова вычисляет контрольную сумму полученных данных и сравнивает результат с контрольной суммой из кадра. Если они совпадают, кадр считается правильным и принимается. Если же контрольные суммы не совпадают, то фиксируется ошибка. Необходимо отметить, что функция исправления ошибок для канального уровня не является обязательной, поэтому в некоторых протоколах этого уровня она отсутствует, например в Ethernet и Frame relay.

Функции канального уровня заключаются в следующем:

- надежная доставка пакета между двумя соседними станциями в сети с произвольной топологией, а также между любыми станциями в сети с типовой топологией;
- проверка доступности разделяемой среды;
- выделение кадров из потока данных, поступающих по сети;
- формирование кадров при отправке данных;
- подсчет и проверка контрольной суммы.

Этот уровень реализуются программно-аппаратно. В протоколах канального уровня, используемых в локальных сетях, заложена определенная структура связей между компьютерами и способы их адресации. Хотя канальный уровень и обеспечивает доставку кадра между любыми двумя узлами локальной сети, он это делает только в сети с совершенно определенной топологией связей, именно той топологией, для которой он был разработан. К таким типовым топологиям, поддерживаемым протоколами канального уровня локальных сетей, относятся общая шина, кольцо и звезда. Примерами протоколов канального уровня являются протоколы Ethernet, Token Ring, FDDI, 100VG-AnyLAN.

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

В глобальных сетях, которые редко обладают регулярной топологией, канальный уровень обеспечивает обмен сообщениями между двумя соседними компьютерами, соединенными индивидуальной линией связи. Примерами протоколов "точка - точка" могут служить широко распространенные протоколы PPP и LAP-B.

Именно так организованы сети X.25. Иногда в глобальных сетях функции канального уровня в чистом виде выделить трудно, так как в одном и том же протоколе они объединяются с функциями сетевого уровня. Примерами такого подхода могут служить протоколы технологий ATM и Frame relay.

В целом канальный уровень представляет собой весьма мощный набор функций по пересылке сообщений между узлами сети.

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

Протоколы канального уровня локальных сетей обеспечивают доставку данных между любыми узлами только в сети с соответствующей типовой топологией, например топологией иерархической звезды. Это жесткое ограничение, которое не позволяет строить сети с развитой структурой, например сети, объединяющие несколько сетей предприятия в единую сеть, или высоконадежные сети, в которых существуют избыточные связи между узлами.

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

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

Сети соединяются между собой специальными устройствами, называемыми маршрутизаторами. Маршрутизатор — это устройство, которое собирает информацию о топологии межсетевых соединений и пересылает пакеты сетевого уровня в сеть назначения. Чтобы передать сообщение от отправителя, находящегося в одной сети, получателю, находящемуся в другой сети, нужно совершить некоторое количество транзитных передач между сетями, или хопов (от слова hop — прыжок), каждый раз выбирая подходящий

маршрут. Таким образом, маршрут представляет собой последовательность маршрутизаторов, через которые проходит пакет.

Свойства сетевого уровня — доставка пакета:

- между любыми двумя узлами сети с произвольной топологией;
- между любыми двумя сетями в составной сети.

При этом, сеть — это совокупность компьютеров, использующих для обмена данными единую сетевую технологию; а маршрут — последовательность прохождения пакетом маршрутизаторов в составной сети.

На рисунке 3 показаны четыре сети, связанные тремя маршрутизаторами. Между узлами A и B данной сети пролегает два маршрута: первый — через маршрутизаторы 1 и 3, а второй — через маршрутизаторы 1, 2 и 3.

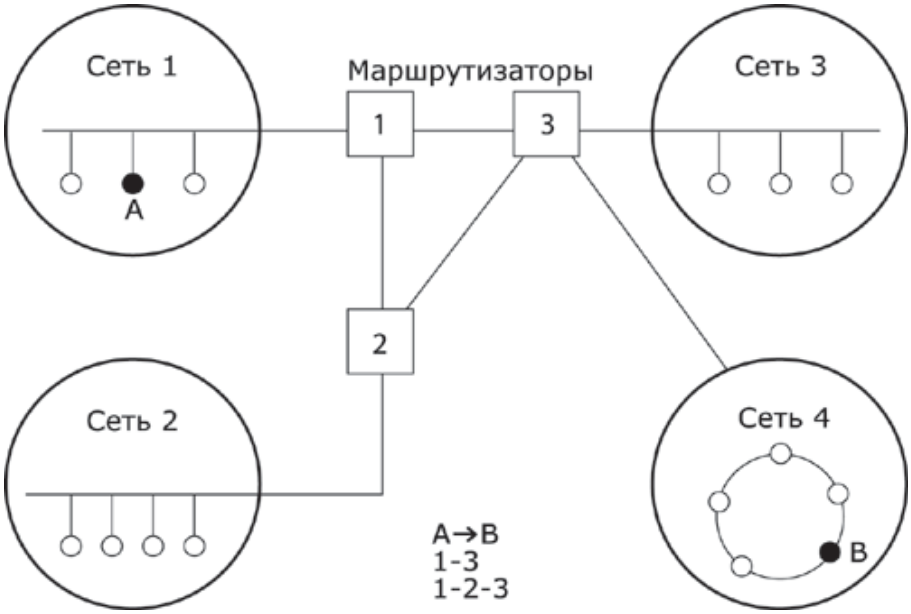


Рисунок 3 - Пример составной сети

В общем случае функции сетевого уровня шире, чем функции передачи сообщений по связям с нестандартной структурой, которые мы рассмотрели на примере объединения нескольких локальных сетей. Сетевой уровень также решает задачи согласования разных технологий, упрощения адресации в крупных сетях и создания надежных и гибких барьеров на пути нежелательного трафика между сетями.

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

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

На сетевом уровне работают протоколы еще одного типа, которые отвечают за отображение адреса узла, используемого на сетевом уровне, в локальный адрес сети. Такие

протоколы часто называют протоколами разрешения адресов — Address Resolution Protocol, ARP. Иногда их относят не к сетевому уровню, а к канальному, хотя тонкости классификации не изменяют сути.

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

Транспортный уровень. На пути от отправителя к получателю пакеты могут быть искажены или утеряны. Хотя некоторые приложения имеют собственные средства обработки ошибок, существуют и такие, которые предпочитают сразу иметь дело с надежным соединением. Работа транспортного уровня заключается в том, чтобы обеспечить приложениям или верхним уровням стека - прикладному и сеансовому - передачу данных с той степенью надежности, которая им требуется. Модель OSI определяет пять классов сервиса, предоставляемых транспортным уровнем. Эти виды сервиса отличаются качеством предоставляемых услуг: срочностью, возможностью восстановления прерванной связи, наличием средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол, а главное - способностью к обнаружению и исправлению ошибок передачи, таких как искажение, потеря и дублирование пакетов.

Выбор класса сервиса транспортного уровня определяется, с одной стороны, тем, в какой степени задача обеспечения надежности решается самими приложениями и протоколами более высоких, чем транспортный, уровней, а с другой стороны, этот выбор зависит от того, насколько надежной является вся система транспортировки данных в сети. Так, например, если качество каналов передачи связи очень высокое, и вероятность возникновения ошибок, не обнаруженных протоколами более низких уровней, невелика, то разумно воспользоваться одним из облегченных сервисов транспортного уровня. Если же транспортные средства изначально очень ненадежны, то целесообразно обратиться к наиболее развитому сервису транспортного уровня, который работает, используя максимум средств для обнаружения и устранения ошибок - с помощью предварительного установления логического соединения, контроля доставки сообщений с помощью контрольных сумм и циклической нумерации пакетов, установления тайм-аутов доставки и т.п.

Т.о. функции транспортного уровня - это обеспечение доставки информации с требуемым качеством между любыми узлами сети:

- разбивка сообщения сеансового уровня на пакеты, их нумерация;
- буферизация принимаемых пакетов;
- упорядочивание прибывающих пакетов;
- адресация прикладных процессов;
- управление потоком.

Как правило, все протоколы, начиная с транспортного уровня и выше, реализуются программными средствами конечных узлов сети - компонентами их сетевых операционных систем. В качестве примера транспортных протоколов можно привести протоколы TCP и UDP стека TCP/IP и протокол SPX стека Novell.

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

Сеансовый уровень. Сеансовый уровень (Session layer) обеспечивает управление диалогом: фиксирует, какая из сторон является активной в настоящий момент, предоставляет средства синхронизации. Последние позволяют вставлять контрольные точки в длинные передачи, чтобы в случае отказа можно было вернуться назад к последней контрольной точке, а не начинать все сначала. На практике немногие приложения используют сеансовый уровень, и он редко реализуется в виде отдельных протоколов, хотя функции этого уровня часто объединяют с функциями прикладного уровня и реализуют в одном протоколе.

Функции сеансового уровня - это управление диалогом объектов прикладного уровня:

- установление способа обмена сообщениями (дуплексный или полудуплексный);
- синхронизация обмена сообщениями;
- организация "контрольных точек" диалога.

Уровень представления. Этот уровень обеспечивает гарантию того, что информация, передаваемая прикладным уровнем, будет понятна прикладному уровню в другой системе. При необходимости уровень представления выполняет преобразование форматов данных в некоторый общий формат представления, а на приеме, соответственно, выполняет обратное преобразование. Таким образом, прикладные уровни могут преодолеть, например, синтаксические различия в представлении данных. На этом уровне может выполняться шифрование и дешифрование данных, благодаря которому секретность обмена данными обеспечивается сразу для всех прикладных сервисов. Примером протокола, работающего на уровне представления, является протокол Secure Socket Layer (SSL), который обеспечивает секретный обмен сообщениями для протоколов прикладного уровня стека TCP/IP.

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

- преобразование данных из внешнего формата во внутренний;
- шифрование и расшифровка данных.

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

Существует очень большое разнообразие протоколов прикладного уровня. Среди них хотелось бы отметить такие как FTP, Telnet, HTTP.

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

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

- идентификация, проверка прав доступа;
- принт- и файл-сервис, почта, удаленный доступ.

Три нижних уровня - физический, канальный и сетевой - являются сетезависимыми, то есть протоколы этих уровней тесно связаны с технической реализацией сети, с используемым коммуникационным оборудованием.

Три верхних уровня - сеансовый, уровень представления и прикладной - ориентированы на приложения и мало зависят от технических особенностей построения сети. На протоколы этих уровней не влияют никакие изменения в топологии сети, замена оборудования или переход на другую сетевую технологию.

Транспортный уровень является промежуточным, он скрывает все детали функционирования нижних уровней от верхних уровней. Это позволяет разрабатывать приложения, независящие от технических средств, непосредственно занимающихся транспортировкой сообщений.

Рисунок 4 показывает уровни модели OSI, на которых работают различные элементы сети. Компьютер, с установленной на нем сетевой ОС, взаимодействует с другим компьютером с помощью протоколов всех семи уровней. Это взаимодействие компьютеры осуществляют через различные коммуникационные устройства. В зависимости от типа, коммуникационное устройство может работать либо только на физическом уровне (повторитель), либо на физическом и канальном (мост и коммутатор), либо на физическом, канальном и сетевом, иногда захватывая и транспортный уровень (маршрутизатор).

Модель OSI представляет хотя и очень важную, но только одну из многих моделей коммуникаций. Эти модели и связанные с ними стеки протоколов могут отличаться

количеством уровней, их функциями, форматами сообщений, сервисами, предоставляемыми на верхних уровнях и прочими параметрами.

Поэтому модель OSI стоит рассматривать, в основном, как опорную базу для классификации и сопоставления протокольных стеков.

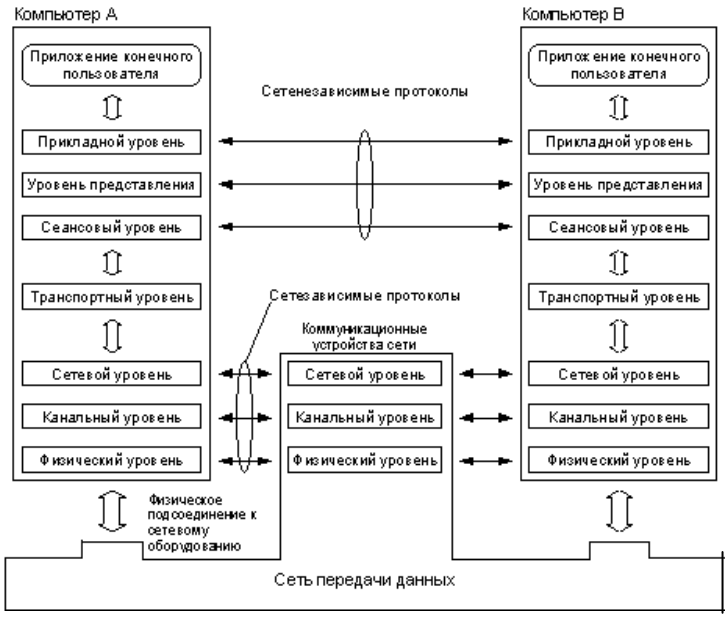


Рисунок 4 - Сетезависимые и сетенезависимые уровни модели OSI

Протокол IP

Архитектуру сетевого уровня удобно рассматривать на примере сетевого протокола IP – самого распространенного в настоящее время, основного протокола сети Интернет. Термин «стек протоколов TCP/IP» означает «набор протоколов, связанных с IP и TCP(протоколом транспортного уровня)».

Архитектура протоколов TCP/IP предназначена для объединенной сети, состоящей из соединенных друг с другом шлюзами отдельных разнородных пакетных подсетей, к которым подключаются разнородные машины.

Каждая из подсетей работает в соответствии со своими специфическими требованиями и имеет свою природу средств связи. Однако предполагается, что каждая подсеть может принять пакет информации (данные с соответствующим сетевым заголовком) и доставить его по указанному адресу в этой конкретной подсети.

Не требуется, чтобы подсеть гарантировала обязательную доставку пакетов и имела надежный сквозной протокол.

Таким образом, две машины, подключенные *к одной подсети*, могут обмениваться пакетами.

Когда необходимо передать пакет между машинами, *подключенными к разным подсетям*, то машина-отправитель посылает пакет в соответствующий шлюз (шлюз подключен к подсети также как обычный узел). Оттуда пакет направляется по определенному маршруту через систему шлюзов и подсетей, пока не достигнет шлюза, подключенного к той же подсети, что и машина-получатель: там пакет направляется к получателю.

Таким образом, адрес получателя должен содержать в себе:

- номер (адрес) подсети;
- номер (адрес) участника (хоста) внутри подсети.

IP адреса представляют собой 32-х разрядные двоичные числа. Для удобства их записывают в виде четырех десятичных чисел, разделенных точками. Каждое число является десятичным эквивалентом

Двоичное	Десятичное

10000000	**128**
11000000	**192**
11100000	**224**
11110000	**240**
11111000	**248**
11111100	**252**
11111110	**254**
11111111	**255**

соответствующего байта адреса (для удобства будем записывать точки и в двоичном изображении).

192.168.200.47 является десятичным эквивалентом двоичного адреса

11000000.10101000.11001000.00101111

Иногда применяют десятичное значение IP-адреса. Его легко вычислить

$$192*256^3+168*256^2+200*256+47=3232286767$$

или с помощью метода Горнера :

$$(((192*256)+168)*256+200)*256+47=3232286767$$

Количество разрядов *адреса подсети* может быть различным и определяется *маской сети*.

Маска сети также является 32-х разрядным двоичным числом. Разряды маски имеют следующий смысл: если разряд маски равен 1, то соответствующий разряд адреса является разрядом адреса подсети, а если 0, то разрядом хоста внутри подсети. Все единичные разряды маски (если они есть) находятся в старшей (левой) части маски, а нулевые (если они есть) – в правой (младшей).

Исходя из вышесказанного, маску часто записывают в виде числа единиц в ней содержащихся.

255.255.248.0 (11111111.11111111.11111000.00000000) – является правильной маской подсети **(/21)**, а **255.255.250.0 (11111111.11111111.11111010.00000000)** – является неправильной, недопустимой.

Нетрудно увидеть, что *максимальный размер подсети* может быть только степенью двойки (двойку надо возвести в степень, равную количеству нулей в маске).

При передаче пакетов используются *правила маршрутизации*, главное из которых звучит так: «Пакеты участникам своей подсети доставляются напрямую, а остальным – по другим правилам маршрутизации».

Таким образом, требуется определить, является ли получатель членом нашей подсети или нет.

1. ***Определение диапазона адресов подсети.***

Определение диапазона адресов подсети можно произвести из определения понятия маски:

- В те разряды, которые относятся к адресу подсети, у всех хостов подсети должны быть *одинаковы*;
- В адреса хостов в подсети могут быть *любыми*.

То есть, если наш адрес **192.168.200.47** и маска равна **/20,** то диапазон можно посчитать:

11000000.10101000.11001000.00101111 –адрес
11111111.11111111.11110000.00000000 –маска
11000000.10101000.1100XXXX.XXXXXXXX –диапазон адресов

где 0,1 – определенные значения
разрядов, X – любое значение,
Что приводит к диапазону адресов:

от **11000000.10101000.11000000.00000000 (192.168.192.0)**
до **11000000.10101000.11001111.11111111**
(192.168.207.255)

Следует учитывать, что некоторые адреса являются запрещенными или служебными и их нельзя использовать для адресов хостов или подсетей. Этоадреса, содержащие:

0. **0** в *первом* или *последнем* байте,
1. **255** в *любом* байте (это широковещательные адреса),
2. **127** в *первом* байте (внутренняя петля – этот адрес имеется в каждом хосте и служит для связывания компонентов сетевого

уровня). Поэтому доступный диапазон адресов будет несколько меньше.

Задания для самостоятельного решения:

1. Какие адреса из приведенного ниже списка являются допустимыми адресами хостов:

0.10.10.10
10.0.10.10
10.10.0.10
10.10.10.10
127.0.127.127
127.0.127.0
255.0.200.1
1.255.0.0

2. Перечислите все допустимые маски.
3. Определите диапазоны адресов подсетей (даны адрес хоста и маска подсети):

10.212.157.12/24
27.31.12.254/31
192.168.0.217/28
10.7.14.14/16

4. Какие из адресов

241.253.169.212
243.253.169.212
242.252.169.212
242.254.169.212
242.253.168.212
242.253.170.212
242.253.169.211
242.253.169.213

будут достигнуты напрямую с хоста **242.254.169.212/21**

5. Посмотрите параметры IP на своем компьютере с помощью команды **ipconfig.** Определите диапазон адресов и размер подсети, в которой Вы находитесь. Попробуйте объяснить, почему выбраны такие сетевые параметры и какие сетевые параметры выбрали бы Вы.

Вопросы для проверки:

1. Чем занимается сетевой уровень?
2. Что такое сеть передачи данных?
3. Какие требования предъявляются к сетевой адресации?
4. Можно ли использовать в качестве сетевого MAC-адрес?

5. Что такое маска подсети,?
6. Какова структура IP-адреса?
7. Чем определяется размер подсети?
8. Как определить диапазон адресов в подсети?
9. Как определить размер подсети?

Сетевой уровень отвечает за возможность доставки пакетов по сети передачи данных – совокупности сегментов сети, объединенных в единую сеть любой сложности посредством узлов связи, в которой имеется возможность достижения из любой точки сети в любую другую.

Архитектуру сетевого уровня удобно рассматривать на примере сетевого протокола IP – самого распространенного в настоящее время, основного протокола сети Интернет. Термин «стек протоколов TCP/IP» означает «набор протоколов, связанных с IP и TCP(протоколом транспортного уровня)».

Архитектура протоколов TCP/IP предназначена для объединенной сети, состоящей из соединенных друг с другом шлюзами отдельных разнородных пакетных подсетей, к которым подключаются разнородные машины.

Каждая из подсетей работает в соответствии со своими специфическими требованиями и имеет свою природу средств связи.

Однако предполагается, что каждая подсеть может принять пакет информации (данные с соответствующим сетевым заголовком) и доставить его по указанному адресу в этой конкретной подсети. Не требуется, чтобы подсеть гарантировала обязательную доставку пакетов и имела надежный сквозной протокол. Таким образом, две машины, подключенные к одной подсети, могут обмениваться пакетами.

Когда необходимо передать пакет между машинами, подключенными к разным подсетям, то машина-отправитель посылает пакет в соответствующий *шлюз* (шлюз подключен к подсети также как обычный узел). *Шлюз* (gateway)– любое сетевое оборудование с несколькими сетевыми интерфейсами и осуществляющее продвижение пакетов между сетями на уровне протоколов сетевого уровня.

Из шлюза пакет направляется по определенному *маршруту* через систему *шлюзов* и подсетей, пока не достигнет шлюза, подключенного к той же подсети, что и машина-получатель; там пакет направляется к получателю.

Таким образом, шлюз выполняет *маршрутизацию* – процедуру нахождения в структуре сети пути достижения получателя (построение пути доставки пакетов).

Если хост подключен к нескольким сетям, он должен иметь несколько сетевых адресов, как минимум столько, сколько каналов к нему подключено.

Маршрутизация производится по *правилам маршрутизации*, сведенным в *таблицу маршрутизации*.

Даже если хост не является шлюзом между подсетями, все равно в нем присутствует таблица маршрутизации, ведь любой хост должен отправлять пакеты напрямую членам своей подсети, через какой-то шлюз другим подсетям и не передавать в сеть пакеты, предназначенные самому себе (заворачивать их по внутренней петле 127.0.0.1).

Таблица маршрутизации имеет следующий вид:

Сетевой адрес	Маска сети	Адрес шлюза	Интерфейс	Метрика
0.0.0.0	0.0.0.0	192.168.200.1	192.168.200.47	30
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.192.0	255.255.240.0	192.168.200.47	192.168.200.47	30
192.168.200.47	255.255.255.255	127.0.0.1	127.0.0.1	30
192.168.200.255	255.255.255.255	192.168.200.47	192.168.200.47	30
224.0.0.0	240.0.0.0	192.168.200.47	192.168.200.47	30
255.255.255.255	255.255.255.255	192.168.200.47	192.168.200.47	1

Сетевой адрес - начальный адрес подсети, порядок достижения которой описывает

правило.
Маска сети - маска подсети, которую описывает правило.
Адрес шлюза- показывает, на какой адрес будут посланы пакеты, идущие в сеть назначения. Если пакеты будут идти напрямую, то указывается собственный адрес (точнее тот адрес того канала, через который будут передаваться пакеты).
Интерфейс - указывает адрес канала, через который будут передаваться пакеты.
Интерфейс *всегда принадлежит хосту*, на котором находится правило.
Метрика - определяет время, за которое пакет достигнет сети назначения.
Правила применяются в порядке уменьшения масок.
Правила с равными масками применяются в порядке увеличения метрики.
Применение правила заключается в определении, принадлежит ли хост назначения сети, указанной в правиле, и если принадлежит, то пакет отправляется на адрес шлюза через интерфейс.

Рассмотрим вышеприведенную таблицу маршрутизации, пересортировав правила:

Сетевой адрес	Маска сети	Адрес шлюза	Интерфейс	Метрика
255.255.255.255	255.255.255.255	192.168.200.47	192.168.200.47	1
192.168.200.47	255.255.255.255	127.0.0.1	127.0.0.1	30
192.168.200.255	255.255.255.255	192.168.200.47	192.168.200.47	30
192.168.192.0	255.255.240.0	192.168.200.47	192.168.200.47	30
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
224.0.0.0	240.0.0.0	192.168.200.47	192.168.200.47	30
0.0.0.0	0.0.0.0	192.168.200.1	192.168.200.47	30
255.255.255.255	255.255.255.255	192.168.200.47	192.168.200.47	1

Обратите внимание на маску сети в первом правиле. Она описывает подсеть размером в 1 хост с адресом **255.255.255.255** – это *широковещательный* адрес. Пакеты будут посылаться на адрес **192.168.200.47** через интерфейс **192.168.200.47**. Это наш адрес, т.е. пакеты будут отправляться напрямую.
192.168.200.255 255.255.255.255 192.168.200.47 192.168.200.47 30
Опять широковещательный адрес. Смотри предыдущий комментарий.
192.168.200.47 255.255.255.255 127.0.0.1 127.0.0.1 30
Опять такая же маска, но адрес нашего хоста. Отправлять будем через внутреннюю петлю.
192.168.192.0 255.255.240.0 192.168.200.47 192.168.200.47 30
А вот и наша подсеть. Отправляем напрямую.
127.0.0.0 255.0.0.0 127.0.0.1 127.0.0.1 1
Все, что начинается со 127, отправляем через внутреннюю петлю.
224.0.0.0 240.0.0.0 192.168.200.47 192.168.200.47 30
Класс D – отправляем напрямую.
0.0.0.0 0.0.0.0 192.168.200.1 192.168.200.47 30

Самое интересное правило. Маска покрывает ВСЕ возможные адреса! Пакеты отправляются через наш интерфейс на адрес **192.168.200.1**. Правило применяется последним, поэтому его можно озвучить так: по всем адресам, которые не подошли по предыдущим правилам, пакеты отправляем на адрес **192.168.200.1**

Такой адрес обычно имеется в любой сети и называется **шлюзом по умолчанию (default gateway)**. Этот адрес скрывает от хостов и пользователей структуру сети и позволяет упростить таблицы маршрутизации и снять нагрузку с хостов, перенеся маршрутизацию на специально выделенные шлюзы – маршрутизаторы.

Нетрудно догадаться, что все адреса в колонке **Адрес шлюза** должны достигаться напрямую, т.е. входить в нашу подсеть.

Для работы с таблицами маршрутизации в составе ОС имеется программа **route.**

Одной из основных задач, стоящих при проектировании сетей, является распределение

по подсетям сетевых адресов из заданного диапазона, т.е. разделение сети на подсети.

При разделении сети на подсети следует учитывать следующие правила:

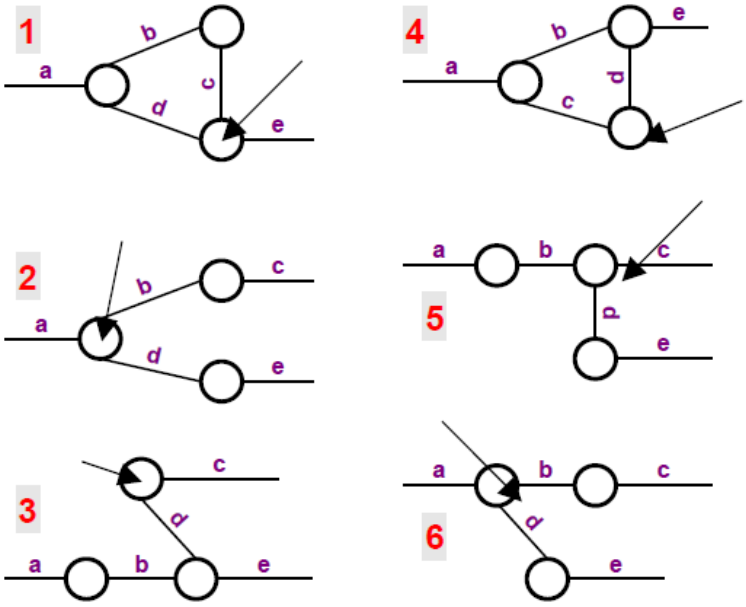
- размер подсетей должен быть *степенью двойки*
- имеются *запрещенные адреса*
- начальный адрес подсети должен быть *кратен ее размеру*

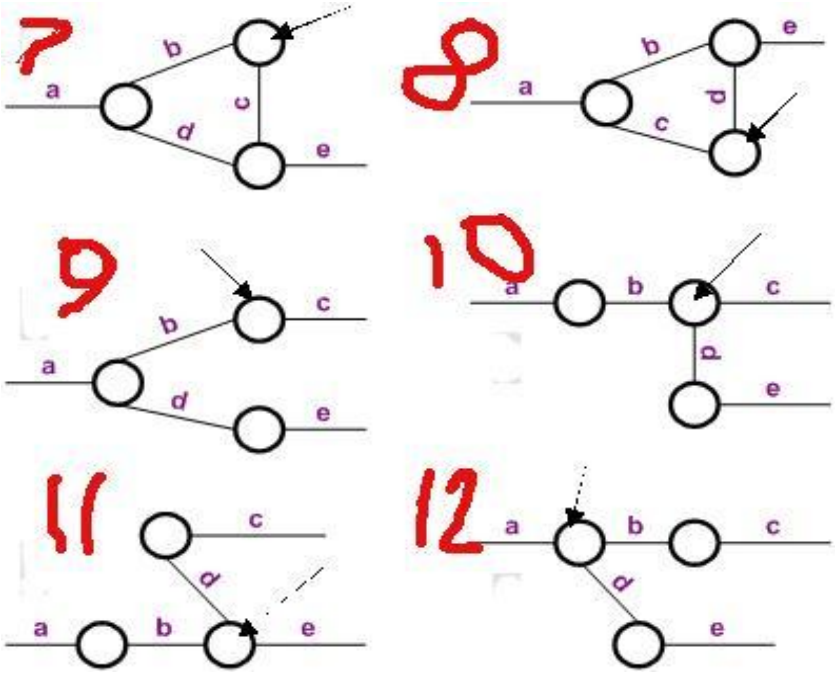
В качестве шлюза по умолчанию можно использовать *любой* узел, но, исходя из увеличения пропускной способности сети и уменьшения времени передачи пакетов, следует в качестве шлюза по умолчанию использовать либо ближайший узел, либо узел, соединенный с максимальным количеством сетей, т.е. следует учитывать *топологию* сети.

Задания для самостоятельного решения:

1. С помощью программы **route print** посмотрите таблицу маршрутизации Вашего компьютера. Объясните все правила.
2. Посмотрите таблицу маршрутизации хоста, имеющего несколько каналов. Объясните все правила.
3. Посмотрите таблицу маршрутизации маршрутизатора. Объясните все правила.
4. Добавьте новое правило в таблицу маршрутизации для сети **192.168.0.0/24** через шлюз в вашей сети с последним байтом в адресе **125** и метрикой **12**
5. Удалите это правило.
6. В соответствии с таблицей и схемами выполните задание на распределение адресов по подсетям (согласно варианта). Постройте таблицы маршрутизации для всех шлюзов и для одного хоста для каждого сегмента.

№ варианта	Количество хостов в подсети					Диапазон адресов	
	a	**b**	**c**	**d**	**e**	**от**	**до**
1	**5**	**10**	**20**	**15**	**50**	**10.0.20.0**	**10.0.20.255**
2	**20**	**15**	**6**	**70**	**25**	**192.168.0.0**	**192.168.0.255**
3	**15**	**25**	**5**	**40**	**5**	**112.38.25.128**	**112.38.25.255**
4	**24**	**32**	**8**	**10**	**2**	**196.13.49.0**	**196.13.49.128**
5	**50**	**16**	**64**	**20**	**15**	**68.76.115.0**	**68.76.115.255**
6	**40**	**6**	**10**	**12**	**5**	**211.3.45.0**	**211.3.45.128**
7	**5**	**10**	**20**	**15**	**50**	**10.0.20.0**	**10.0.20.255**
8	**16**	**12**	**8**	**60**	**20**	**92.190.0.0**	**92.190.0.255**
9	**30**	**15**	**7**	**20**	**8**	**34.40.25.128**	**34.40.25.255**
10	**34**	**22**	**15**	**3**	**2**	**76.17.46.0**	**76.17.46.128**
11	**30**	**26**	**54**	**30**	**15**	**8.71.15.0**	**8.71.15.255**
12	**30**	**16**	**6**	**16**	**5**	**71.5.55.0**	**71.5.55.128**





7. Разделите сеть, состоящую из трех сегментов, имеющую диапазон адресов 192.168.0.32

– 192.168.0.159 на подсети, содержащие 64, 20 и 44 хостов (включая шлюзы).

Вопросы для проверки:

1. Сколько адресов может иметь хост?
2. Может ли у хоста быть прописано несколько шлюзов и почему?
3. Может ли у хоста быть прописано несколько шлюзов по умолчанию и почему?
4. Чем отличаются таблицы у разных классов сетевых устройств и почему?
5. Почему начальный адрес подсети должен быть кратен ее размеру?
6. Чем Вы руководствовались при выборе шлюзов по умолчанию?
7. Может ли физический сегмент сети содержать несколько сетевых подсетей?

2.2 Лабораторная работа №2 (2 часа)

Тема: «Запоминающие устройства ЭВМ»

2.2.1 Цель работы: изучить основные запоминающие устройства ЭВМ.

2.2.2 Задачи работы:

1. рассмотреть классификацию запоминающих устройств;
2. изучить основные запоминающие устройства ЭВМ.

2.2.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. персональный компьютер, включенный в сеть IP, Microsoft Windows.

2.2.4 Описание (ход) работы:

Запоминающие устройства классифицируют:

1. По типу запоминающих элементов (полупроводниковые, магнитные, конденсаторные, оптоэлектронные, голографические, криогенные).

2. По функциональному назначению (оперативные (ОЗУ), буферные (БЗУ), сверхоперативные (СОЗУ), внешние (ВЗУ), постоянные (ПЗУ)).

3. По способу организации обращения (с последовательным поиском, с прямым доступом, адресные, ассоциативные, стековые, магазинные).

4. По характеру считывания (с разрушением или без разрушения информации).

5. По способу хранения (статические или динамические).

6. По способу организации (однокоординатные, двухкоординатные, трехкоординатные, двух/трехкоординатные).

ПАМЯТЬ ЭВМ - совокупность всех запоминающих устройств, входящих в состав ЭВМ. Обычно в состав ЭВМ входит несколько различных типов ЗУ.

Производительность и вычислительные возможности ЭВМ в значительной степени определяются составом и характеристиками ее ЗУ.

Основными операциями в памяти в общем случае являются занесение информации в память - запись и выборка информации из памяти - считывание. Обе эти операции называются обращением к памяти или, подробнее, обращением при считывании и обращением при записи.

При обращении к памяти производится считывание или запись некоторой единицы данных - различной для устройств разного типа. Такой единицей может быть бит, байт, машинное слово или блок данных.

Важнейшими характеристиками отдельных устройств памяти являются емкость памяти, удельная емкость, быстродействие.

Емкость памяти определяется максимальным количеством данных, которые могут в ней храниться. Емкость измеряется в двоичных единицах (битах), машинных словах, но большей частью в байтах.

Удельная емкость есть отношение емкости ЗУ к его физическому объему.

Быстродействие памяти определяется продолжительностью операций обращения, т.е. временем, затрачиваемым на поиск единицы информации в памяти и на ее считывание, или временем на поиск места в памяти, предназначенного для хранения данной единицы информации, и на ее запись.

ВЗУ (внешнее запоминающее устройство) - запоминающее устройство, предназначенное для длительного хранения массивов информации и обмена ими с ОЗУ. Обычно строятся на базе магнитных носителей информации. Само название этого класса устройств имеет исторический характер и произошло от больших ЭВМ, в которых все ВЗУ, как более медленные и громоздкие, размещались в собственном корпусе, а не в корпусе основного модуля.

Внутренняя память ЭВМ организуется как взаимосвязанная совокупность нескольких типов ЗУ. В ее состав, кроме ОЗУ, могут входить следующие типы ЗУ:

Постоянное запоминающее устройство (пзу) - запоминающее устройство, из которого может производиться только выдача хранящейся в нем информации. Занесение информации в ПЗУ производится при его изготовлении.

Полупостоянное (программируемое) зу (ппзу) - ЗУ, в котором информация может обновляться с помощью специальной аппаратуры перед режимом автоматической работы ЭВМ. Если возможно многократное обновление информации, то иногда такое ППЗУ называют репрограммируемым (РППЗУ).

Буферное запоминающее устройство (бзу) - запоминающее устройство, предназначенное для промежуточного хранения информации при обмене данными между устройствами ЭВМ, работающими с различными скоростями. Конструктивно оно может быть частью любого из функциональных устройств.

Местная память (сверхоперативное ЗУ, СОЗУ) - буферное запоминающее устройство, включаемое между ОЗУ и процессором или каналами. Различают местную память процессора и местную память каналов.

СТЕК (магазин) - специально организованное ОЗУ, блок хранения которого состоит из регистров, соединенных друг с другом в цепочку, по которой их содержимое при обращении к ЗУ передается (сдвигается) в прямом или обратном направлении.

Кеш-память - разновидность стека, в котором хранятся копии некоторых команд из ОЗУ.

ВИДЕОПАМЯТЬ - область ОЗУ ЭВМ, в которой размещены данные, видимые на экране дисплея.

2.3 Лабораторная работа № 3 (2 часа)
Тема: «Протоколы и алгоритмы маршрутизации»
2.3.1 Цель работы: получить сведения о маршрутизации и научиться добавлять маршруты в таблицу маршрутизации.
2.3.2 Задачи работы:
1. Научиться работать с таблицей маршрутизацией.
2.3.3 Перечень приборов, материалов, используемых в лабораторной работе:
1. Аппаратные: компьютер с установленной ОС Windows.
2. Программные: Приложения ВМ: VirtualBox; Виртуальные машины: VM-1.
2.3.4 Описание (ход) работы:

В сетях, основанных на протоколе IP, *концепция маршрутизации* является одной из важных. Она создает или разбивает сеть. Неправильная конфигурация маршрутизации способна вывести из строя сеть.
Маршрутизация – технология определения пути доставки (маршрута) пакетов. Основные принципы маршрутизации:

1. Каждая операционная система, поддерживающая стек **TCP/IP**, имеет маршрутизатор и таблицу маршрутизации.
2. Таблица маршрутизации используется только тогда, когда определяется, как доставлять пакеты.
3. Маршрутизация должна быть сконфигурирована корректно на обоих концах связи и на каждом участке между ними.

Для определения пути доставки пакета используется *таблица маршрутизации*. Пример таблицы маршрутизации можно получить командой **route** с параметром *print*.

Активные маршруты:

Сетевой адрес	Маска сети	Адрес шлюза	Интерфейс	Метрика
Network Destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.4.1	192.168.4.7	1
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.4.0	255.255.255.0	192.168.4.7	192.168.4.7	1
192.168.4.7	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.4.255	255.255.255.255	192.168.4.7	192.168.4.7	1
224.0.0.0	224.0.0.0	192.168.4.7	192.168.4.7	1
255.255.255.255	255.255.255.255	192.168.4.7	192.168.4.7	1
Основной шлюз:	192.168.1.1			

Рисунок 1. Пример таблицы маршрутизации

В общем случае для маршрутизации используется следующий *алгоритм*. Из пакета извлекается IP-адрес назначения пакета и производится попытка сопоставить его с адресом назначения (*Сетевой адрес*) каждого элемента таблицы маршрутизации пока не найдется наилучшее совпадение. Если совпадений не найдено, то пакет удаляется и отправителю пакета может отправиться сообщение об ошибке. Сравнение производится с тремя порциями информации: **Сетевой адрес (*Network Destination*), Маска сети (*Netmask*) и IP-адрес назначения пакета.**

В основном, производится побитная операция **AND** между **IP-адресом получателя** и **Маской сетии (*Netmask*)**: если полученное значение равно **Сетевому адресу (*Network Destination*)**, то считается, что совпадение найдено.

Пример 1. Необходимо проверить почту на сервере, чей адрес **192.168.4.100** (используется таблица маршрутизации приведенная ранее). Необходимо выполнить побитную операцию **AND** над **IP-адресом получателя пакетов** и **сетевыми масками (*Netmask*)** из таблицы маршрутизации. Эта операция производится над всем масками из таблицы маршрутизации. Но в рассматриваемом примере только 3-я строка наиболее походит.

	1-й октет								2-й октет								3-й октет								4-й октет							
биты	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ID-сети																													ID-узла		
IP-адрес	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1	0	0
десятичная запись	**192**								**168**								**4**								**100**							
Маска	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
десятичная запись	**255**								**255**								**255**								**0**							
Результат операции AND	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
десятичная запись	**192**								**168**								**4**								**0**							

Рисунок 2. Пример определения маршрута доставки пакетов

Как видно из приведенной таблицы, результат побитной операции **AND** совпадает с 3-й строкой таблицы маршрутизации (Рисунок 2). Следовательно, пакет отправится по указанному маршруту через интерфейс**192.168.4.7.**

Следует отметить, что указанный в примере IP-адрес после выполнения побитной операции **AND** над масками совпадет больше чем с одной строкой маршрутизации. Для избежания таких случаев используется*приоритет маршрутов*. Система ищет более точное совпадение адреса с маской (255.255.255.255 более точна, чем 255.255.255.0, которая в свою очередь, более точна, чем 0.0.0.0). Маршрут с сетевым адресом 0.0.0.0 и маской 0.0.0.0 является *маршрутом по умолчанию*. Так как этот маршрут подходит к любому адресу назначения, он описывает маршрут, который используется, если не найден более подходящий. Обычно этот маршрут используется для пересылки пакетов провайдеру Интернет-услуг, при подключении к Интернету.

Для работы с таблицей маршрутизации используется стандартная утилита **ROUTE,** которая выводит на экран и изменяет записи в локальной таблице IP-маршрутизации.

Запущенная без параметров, команда **route** выводит справку.

Параметр	**Описание**
add	Добавление маршрута
change	Изменение существующего маршрута
delete	Удаление маршрута или маршрутов
print	Печать маршрута или маршрутов

*Таблица 1. Назначение параметров команды **route***

Пример 2. Добавление маршрута.

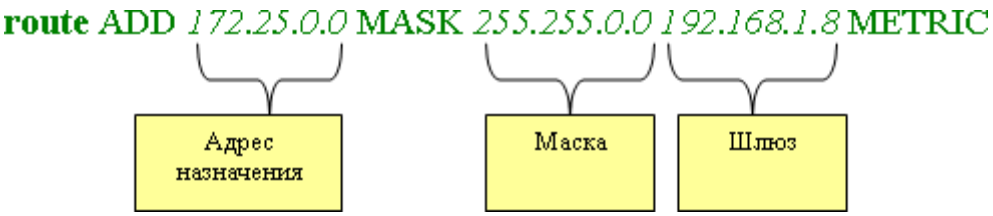


Рисунок 3. СТрока для добавление маршрута

Задание 1. Создайте таблицу для облегчения определения маршрутов.

1. Откройте **табличный процессор** и сформируйте таблицу по следующему шаблону:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG
1		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
2	**IP-адрес**	192								168								252								56							
3	двоичная запись	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0
4	**Маска**	255								255								255								7							
5	двоичная запись	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1
6	**Операция AND**	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
7	**Адрес назначения**																																

Рисунок 4. Образец оформления таблицы

2. Введите в диапазон ячеек **Z3:AG3** формулы для перевода числа в десятичной системе счисления из ячейки **Z2** в двоичную форму (в соответствии с таблицей).

Таблица 2. Формулы для перевода в двоичную систему счисления

Имя Ячейки	**Формула**
AG3	=Z2-2*INT(Z2/2)
AF3	=INT(Z2/2)-2*INT(INT(Z2/2)/2)
AE3	=INT(INT(Z2/2)/2)-2*INT(INT(INT(Z2/2)/2)/2)
AD3	=INT(INT(INT(Z2/2)/2)/2)-2*INT(INT(INT(INT(Z2/2)/2)/2)/2)
AC3	=INT(INT(INT(INT(Z2/2)/2)/2)/2)- 2*INT(INT(INT(INT(INT(Z2/2)/2)/2)/2)/2)
AB3	=INT(INT(INT(INT(INT(Z2/2)/2)/2)/2)/2)- 2*INT(INT(INT(INT(INT(INT(Z2/2)/2)/2)/2)/2)/2)
AA3	=INT(INT(INT(INT(INT(INT(Z2/2)/2)/2)/2)/2)/2)- 2*INT(INT(INT(INT(INT(INT(INT(Z2/2)/2)/2)/2)/2)/2)/2)
Z3	=INT(INT(INT(INT(INT(INT(INT(Z2/2)/2)/2)/2)/2)/2)/2)

3. Аналогично введите формулы для преобразования чисел из десятичной системы счисления в двоичную для ячеек **R2,J2,B2**.
4. Аналогично введите формулы для преобразования маски подсети в двоичную систему счисления.
5. Введите формулы для побитной операции **AND** над **IP-адресом** и **маской (*Netmask*)**:
   - введите в ячейку **AG6** формулу **=AND(AG3;AG5)**;
   - скопируйте введенную формулу в диапазон ячеек **B6:AF6**.

6. Введите в ячейку **Z7** формулу для преобразования 4-го октета маски в десятичную систему счисления - **=AG6*2^AL1+AF6*2^AF1+AE6*2^AE1+AD6*2^AD1+AC6*2^AC1+AB6*2^AB1+AA6*2^AA1+Z6*2^Z1**.
7. Аналогично введите формулы для ячеек **R7, J7, B8**.
8. Сохраните файл в своем каталоге с именем ***ROUTE***.

Задание 2. Создайте новый маршрут для вашего компьютера и проследите его.

1. Запустите виртуальную машину **VM-1** и загрузите ОС **Windows**.
2. Откройте **консоль (*Пуск/Программы/Стандартные/Командная строка*)**.
3. Определите IP-адрес вашего компьютера с помощью утилиты **ipconfig**.
4. Просмотрите таблицу маршрутизации на вашем компьютере:

- выведите справку по команде **route** (для этого необходимо ввести команду и нажать клавишу ***ENTER***);

```
route
```

- выведите таблицу маршрутизации командой **route** с параметром *PRINT*:

```
route PRINT
```

- запомните маршрут по умолчанию (первая строка).

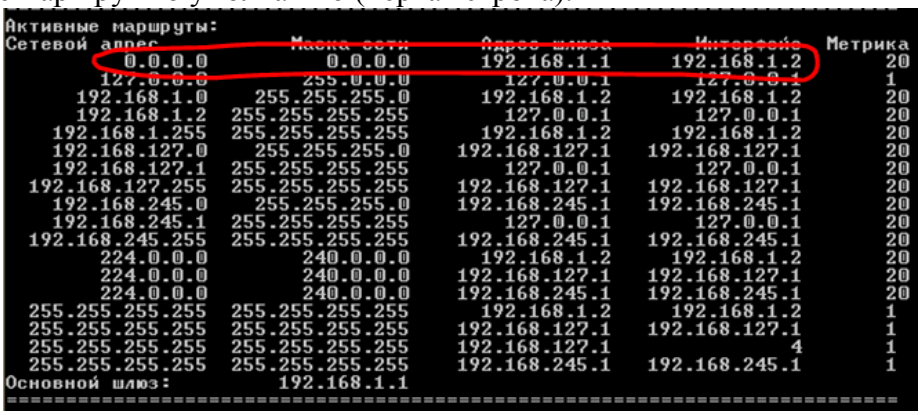


Рисунок 5.Пример вывода программы **ROUTE**

5. Проследите работу маршрутизатора с помощью утилиты *TRACERT*, отправив пакеты на узел **www.opennet.ru**. Введите:

tracert www.opennet.ru

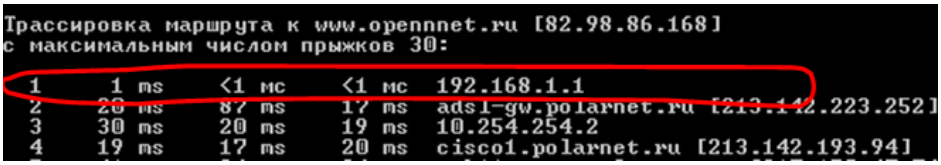


Рисунок 6. Пример вывода программы **TRACERT**

6. *Следует отметить, что пакеты на указанный сайт отправляются через один шлюз (**192.168.1.1**), который видно в первых строках вывода программ **ROUTE** и **TRACERT**.*
7. Добавьте в таблицу маршрутизации компьютера строку для пересылки пакетов в сеть **172.21.0.0** (маска **255.255.0.0**) через сетевой интерфейс компьютера. Введите:

```
route add 172.21.0.0 mask 255.255.0.0 192.168.1.4 METRIC 3
```

8. Проверьте работу внесенных вами изменений с помощью утилиты *TRACERT*.

Самостоятельные задания.

1. Сформируйте маски подсети таким образом, чтобы получались сети, в которых количество уникальных адресов составляют 256, 2048, 32768.
2. Определите маршруты для пакетов в соответствии с таблицей маршрутизации, приведенной в теоретической части лабораторной работы. Результат оформите в виде таблицы и сохраните в своей папке.

Таблица маршрутизации		
IP-адреса пакетов	Адрес шлюза	Интерфейс
10.1.1.1		
192.168.4.121		
127.13.13.210		
192.168.5.121		

2.4 Лабораторная работа №4 (2 часа)

Тема: «Устройства ввода вывода информации в ЭВМ»

2.4.1 Цель работы: научиться работать с устройствами ввода-вывода.

2.4.2 Задачи работы:

1. рассмотреть основные устройства ввода-вывода.

2.4.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. персональный компьютер, включенный в сеть IP, Microsoft Windows.

2.4.4 Описание (ход) работы:

Устройства ввода и вывода - устройства взаимодействия компьютера с внешним миром: с пользователями или другими компьютерами. Устройства ввода позволяют вводить информацию в компьютер для дальнейшего хранения и обработки, а устройства вывода - получать информацию из компьютера.

Устройства ввода и вывода относятся к периферийным (дополнительным) устройствам.

Периферийные устройства - это все устройства компьютера, за исключением процессора и внутренней памяти.

Классификация периферийных устройств по месту расположения (относительного системного блока настольного компьютера или корпуса ноутбука):

внутренние - находятся внутри системного блока\корпуса ноутбука: жесткий диск (винчестер), встроенный дисковод (привод дисков);

внешние - подключаются к компьютеру через порты ввода-вывода: мышь, принтер и т.д.

По другому определению, периферийными устройствами называют устройства, не входящие в системный блок компьютера.

Устройства ввода и вывода разделяются на:

устройства ввода,

устройства вывода,

устройства ввода-вывода.

Устройства ввода данных

Классификация по типу вводимой информации:

устройства ввода текста: клавиатура;

устройства ввода графической информации:

сканер,

цифровые фото- и видеокамера,

веб камера - цифровая фото- или видеокамера маленького размера, которая делает фото или записывает видео в реальном времени для дальнейшей их передачи по сети Интернет;

графический планшет (дигитайзер) - для ввода чертежей, графиков и планов с помощью специального карандаша, которым водят по экрану планшета;

устройства ввода звука: микрофон;

Устройства-манипуляторы (преобразуют движение руки в управляющую информацию для компьютера):

несенсорные:

мышь,

трекбол - устройство в виде шарика, управляется вращением рукой;

трекпойнт (Pointing stick) - джойстик очень маленького размера (5 мм) с шершавой вершиной, который расположен между клавишами клавиатуры, управляется нажатием пальца;

игровые манипуляторы: джойстик, педаль, руль, танцевальная платформа, игровой пульт (геймпад, джойпад);

сенсорные:

тачпад (сенсорный коврик) - прямоугольная площадка с двумя кнопками, управляется движением пальца и нажатием на кнопки, используется в ноутбуках,

сенсорный экран - экран, который реагирует на прикосновение пальца или стилуса (палочка со специальным наконечником), используется в планшетных персональных компьютерах;

графический планшет (дигитайзер) - для ввода чертежей, схем и планов с помощью специального карандаша, которым водят по экрану планшета,

световое перо - устройство в виде ручки, ввод данных приконовением или проведением линий по экрану ЭЛТ-монитора (монитора на основе электронно-лучевой трубки). Сейчас световое перо не используется.

Устройства вывода данных

Классификация по типу выводимой информации:

устройства вывода графической и текстовой информации:

монитор - для вывода на дисплей (экран монитора),

проектор - для вывода на большой экран,

устройства для вывода на печать:

принтер - для вывода информации на бумагу, а также на поверхность дисков;

широкоформатный принтер ("широкий" принтер) - для вывода на листах форматов: A0, A1, A2 и A3,

плоттер (графопостроитель) - для вывода векторных изображений (различных чертежей и схем) на бумаге, картоне, кальке;

каттер (режущий плоттер) - вырезает изображения из пленки, картона по заданному контуру;

устройства вывода (воспроизведения) звука :

наушники,

колонки и акустические системы (динамик, усилитель),

встроенный динамик (PC speaker; Beeper) - для подачи звукового сигнала в случае возникновения ошибки.

Устройства ввода-вывода:

жесткий диск (винчестер) (входящий в него дисковод) - для ввода-вывода информации на жесткие пластины жесткого диска;

флэшка (флешка или USB-флеш-накопитель) - для ввода-вывода информации на микросхему памяти флэшки

дисководод оптических дисков - для ввода-вывода информации на оптические диски,

дисководод гибких дисков - для ввода-вывода информации на дискеты,

стример - для ввода-вывода информации на картриджи (ленточные носители);

кардридер - для ввода-вывода информации на карту памяти;

многофункциональное устройство (МФУ) - копировальный аппарат с дополнительными функциями принтера (вывод данных) и сканера (ввод данных)

модем (телефонный) - для связи компьютеров через телефонную сеть;

сетевая плата (сетевая карта или сетвой адаптер) - для подключения персонального компьютера к сети и организации взаимодействия с другими устройствами сети (обмен информацией по сети).

2.5 Лабораторная работа №5 (2 часа)

Тема: **«**Виды архитектур ЛВС**»**

2.5.1 Цель работы: научиться работать с различными видами архитектур.

2.5.2 Задачи работы:

1. рассмотреть основные виды архитектур.

2.5.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. персональный компьютер, включенный в сеть IP, Microsoft Windows.

2.5.4 Описание (ход) работы:

Вычислительная сеть (ВС) – это сложный комплекс взаимосвязанных и согласованно функционирующих аппаратных и программных компонентов. Аппаратными компонентами локальной сети являются компьютеры и различное коммуникационное оборудование (кабельные системы, концентраторы и т. д.). Программными компонентами ВС являются операционные системы (ОС) и сетевые приложения.

Компоновкой сети называется процесс составления аппаратных компонентов с целью достижения нужного результата.

В зависимости от того, как распределены функции между компьютерами сети, они могут выступать в трех разных ролях:

1. Компьютер, занимающийся исключительно обслуживанием запросов других компьютеров, играет роль выделенного сервера сети (рис. 1.4).
2. Компьютер, обращающийся с запросами к ресурсам другой машины, играет роль узла-клиента (рис. 1.5).
3. Компьютер, совмещающий функции клиента и сервера, является одноранговым узлом (рис. 1.6).

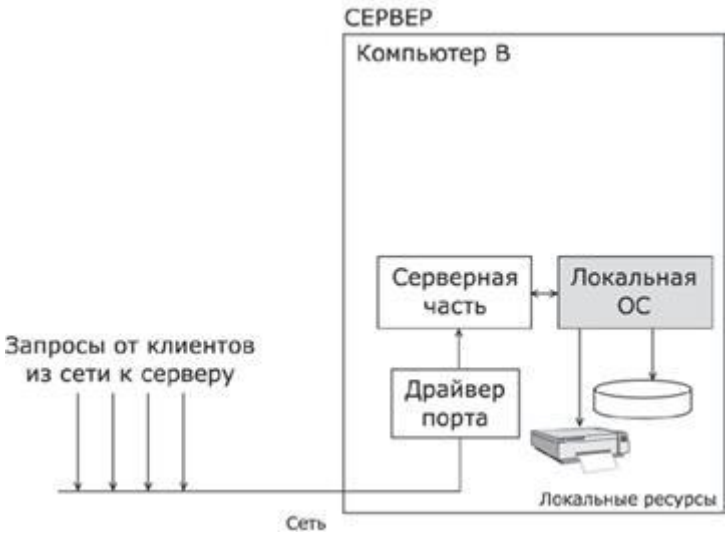


Рис. 1.4. Компьютер - выделенный сервер сети

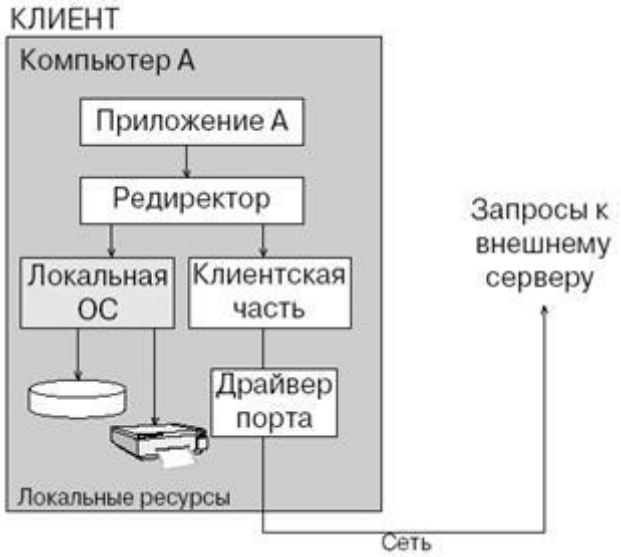


Рис. 1.5. Компьютер в роли узла-клиента

Очевидно, что сеть не может состоять только из клиентских или только из серверных узлов.

Сеть может быть построена по одной из трех схем:

- сеть на основе одноранговых узлов – одноранговая сеть;
- сеть на основе клиентов и серверов – сеть с выделенными серверами;
- сеть, включающая узлы всех типов – гибридная сеть.

Каждая из этих схем имеет свои достоинства и недостатки, определяющие их области применения.

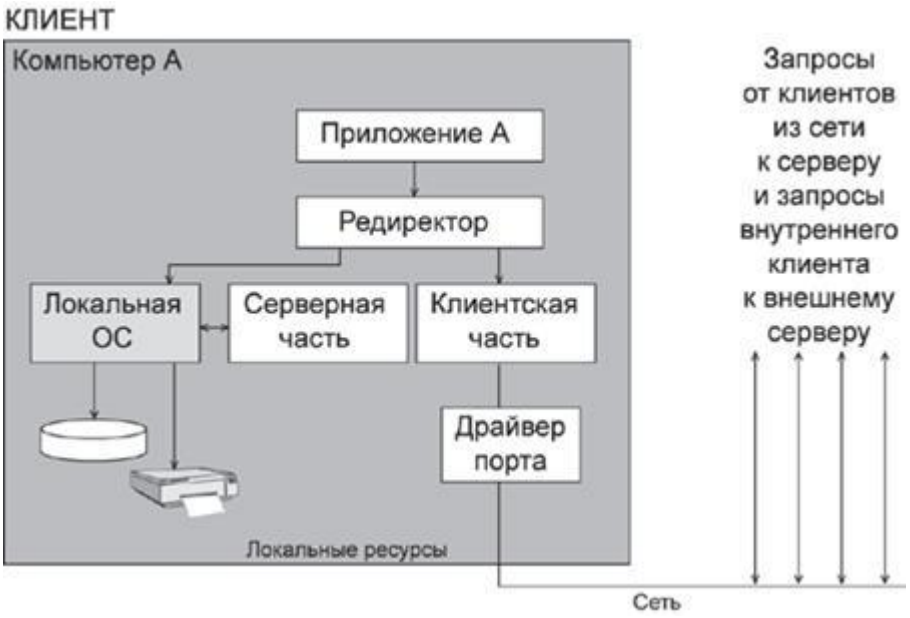


Рис. 1.6. Компьютер - одноранговый узел

В одноранговых сетях один и тот же ПК может быть и сервером, и клиентом, в том числе и клиентом своего клиента. В иерархических сетях разделяемые ресурсы хранятся только на сервере, сам сервер может быть клиентом только другого сервера более высокого уровня иерархии.

При этом каждый из серверов может быть реализован как на отдельном компьютере, так и в небольших по объему ЛВС, быть совмещенным на одном компьютере с каким-либо другим сервером.

Существуют и комбинированные сети, сочетающие лучшие качества одноранговых сетей и сетей на основе сервера. Многие администраторы считают, что такая сеть наиболее полно удовлетворяет их запросы.

Архитектура сети определяет основные элементы сети, характеризует ее общую логическую организацию, техническое обеспечение, программное обеспечение, описывает методы кодирования. Архитектура также определяет принципы функционирования и интерфейс пользователя.

Далее будет рассмотрено три вида архитектур:

- архитектура терминал-главный компьютер;
- одноранговая архитектура;
- архитектура клиент-сервер.

Архитектура терминал-главный компьютер

Архитектура терминал-главный компьютер (terminal-host computer architecture) – это концепция информационной сети, в которой вся обработка данных осуществляется одним или группой главных компьютеров.

Рассматриваемая архитектура предполагает два типа оборудования:

- главный компьютер, где осуществляется управление сетью, хранение и обработка данных;
- терминалы, предназначенные для передачи главному компьютеру команд на организацию сеансов и выполнения заданий, ввода данных для выполнения заданий и получения результатов.

Главный компьютер через МПД взаимодействуют с терминалами, как представлено на рис. 1.7.

Классический пример архитектуры сети с главными компьютерами – системная сетевая архитектура (System Network Architecture – SNA).

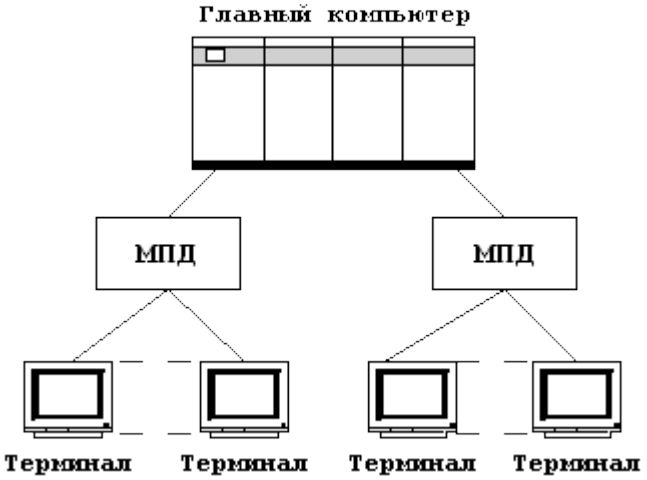


Рис. 1.7. Архитектура терминал-главный компьютер

Одноранговая архитектура

Одноранговая архитектура (peer-to-peer architecture) – это концепция информационной сети, в которой ее ресурсы рассредоточены по всем системам. Данная архитектура характеризуется тем, что в ней все системы равноправны.

К одноранговым сетям относятся малые сети, где любая рабочая станция может выполнять одновременно функции файлового сервера и рабочей станции. В одноранговых ЛВС дисковое пространство и файлы на любом компьютере могут быть общими. Чтобы ресурс стал общим, его необходимо отдать в общее пользование, используя службы удаленного доступа сетевых одноранговых операционных систем. В зависимости от того, как будет установлена защита данных, другие пользователи смогут пользоваться файлами сразу же после их создания. Одноранговые ЛВС достаточно хороши только для небольших рабочих групп.

Одноранговые ЛВС являются наиболее легким и дешевым типом сетей для установки. При соединении компьютеров, пользователи могут предоставлять ресурсы и информацию в совместное пользование.

Одноранговые сети имеют следующие преимущества:

- они легки в установке и настройке;
- отдельные ПК не зависят от выделенного сервера;
- пользователи в состоянии контролировать свои ресурсы;
- малая стоимость и легкая эксплуатация;
- минимум оборудования и программного обеспечения;
- нет необходимости в администраторе;
- хорошо подходят для сетей с количеством пользователей, не превышающим десяти.

Проблемой одноранговой архитектуры является ситуация, когда компьютеры отключаются от сети. В этих случаях из сети исчезают виды сервиса, которые они предоставляли. Сетевую безопасность одновременно можно применить только к одному ресурсу, и пользователь должен помнить столько паролей, сколько сетевых ресурсов. При получении доступа к разделяемому ресурсу ощущается падение производительности компьютера. Существенным недостатком одноранговых сетей является отсутствие централизованного администрирования.

Использование одноранговой архитектуры не исключает применения в той же сети также архитектуры терминал-главный компьютер или архитектуры клиент-сервер.

Архитектура клиент-сервер

Архитектура клиент-сервер (client-server architecture) – это концепция информационной сети, в которой основная часть ее ресурсов сосредоточена в серверах, обслуживающих своих клиентов (рис. 1.8). Рассматриваемая архитектура определяет два типа компонентов: серверы и клиенты.

Сервер – это объект, предоставляющий сервис другим объектам сети по их запросам. Сервис – это процесс обслуживания клиентов.

Сервер работает по заданиям клиентов и управляет выполнением их заданий. После выполнения каждого задания сервер посылает полученные результаты клиенту, пославшему это задание.

Сервисная функция в архитектуре клиент-сервер описывается комплексом прикладных программ, в соответствии с которым выполняются разнообразные прикладные процессы.

Рис. 1.8. Архитектура клиент – сервер

Процесс, который вызывает сервисную функцию с помощью определенных операций, называется клиентом. Им может быть программа или пользователь. На рис. 1.9 приведен перечень сервисов в архитектуре клиент-сервер.

Клиенты – это рабочие станции, которые используют ресурсы сервера и предоставляют удобные интерфейсы пользователя. Интерфейсы пользователя (рис. 1.9) это процедуры взаимодействия пользователя с системой или сетью.

В сетях с выделенным файловым сервером на выделенном автономном ПК устанавливается серверная сетевая операционная система. Этот ПК становится сервером. ПО, установленное на рабочей станции, позволяет ей обмениваться данными с сервером. Наиболее распространенные сетевые операционная системы:

- NetWare фирмы Novel;
- Windows NT фирмы Microsoft;
- UNIX фирмы AT&T;
- Linux.

Помимо сетевой операционной системы необходимы сетевые прикладные программы, реализующие преимущества, предоставляемые сетью.

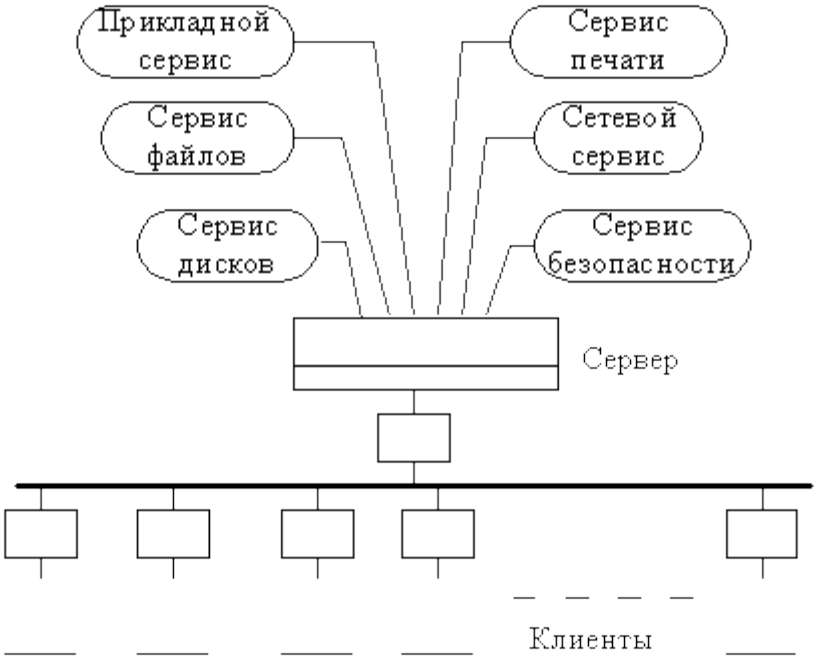


Рис. 1.9. Модель клиент-сервер

Круг задач, которые выполняют серверы в иерархических сетях, многообразен и сложен. Чтобы приспособиться к возрастающим потребностям пользователей, серверы в ЛВС стали специализированными. Так, например, в операционной системе Windows NT Server существуют различные типы серверов:

1. Файл-серверы и принт-серверы. Они управляют доступом пользователей к файлам и принтерам. Так, например, для работы с текстовым документом вы прежде всего запускаете на своем компьютере (PC) текстовыйпроцессор. Далее требуемый документ текстового процессора, хранящийся на файл-сервере, загружается в память PC, и таким образом Вы можете работать с этим документом на PC. Другими словами, файл-сервер предназначен для хранения файлов и данных.
2. Серверы приложений (в том числе сервер баз данных (БД), WEB-сервер). На них выполняются прикладные части клиент серверных приложений (программ). Эти серверы принципиально отличаются от файл-серверов тем, что при работе с файл-сервером нужный файл или данные целиком копируются на запрашивающий PC, а при работе с сервером приложений на PC пересылаются только результаты запроса. Например, по запросу можно получить только список работников, родившихся в сентябре, не загружая при этом в свою PC всю базу данных персонала.
3. Почтовые серверы управляют передачей электронных сообщений между пользователями сети.
4. Факс-серверы управляют потоком входящих и исходящих факсимильных сообщений через один или несколько факс-модемов.
5. Коммуникационные серверы управляют потоком данных и почтовых сообщений между данной ЛВС и другими сетями или удаленными пользователями через модем и телефонную линию. Они же обеспечивают доступ к Internet.
6. Сервер служб каталогов предназначен для поиска, хранения и защиты информации в сети. Windows NT Server объединяет PC в логические группы-домены, система защиты которых наделяет пользователей различными правами доступа к любому сетевому ресурсу.

Клиент является инициатором и использует электронную почту или другие сервисы сервера. В этом процессе клиент запрашивает вид обслуживания, устанавливает сеанс, получает нужные ему результаты и сообщает об окончании работы.

Сети на базе серверов имеют лучшие характеристики и повышенную надежность. Сервер владеет главными ресурсами сети, к которым обращаются остальные рабочие станции.

В современной клиент-серверной архитектуре выделяется четыре группы объектов: клиенты, серверы, данные и сетевые службы. Клиенты располагаются в системах на рабочих местах пользователей. Данные в основном хранятся в серверах. Сетевые службы являются совместно используемыми серверами и данными. Кроме того службы управляют процедурами обработки данных.

Сети клиент-серверной архитектуры имеют следующие преимущества:

- позволяют организовывать сети с большим количеством рабочих станций;
- обеспечивают централизованное управление учетными записями пользователей, безопасностью и доступом, что упрощает сетевое администрирование;
- эффективный доступ к сетевым ресурсам;
- пользователю нужен один пароль для входа в сеть и для получения доступа ко всем ресурсам, на которые распространяются права пользователя.

Наряду с преимуществами сети клиент-серверной архитектуры имеют и ряд недостатков:

- неисправность сервера может сделать сеть неработоспособной;
- требуют квалифицированного персонала для администрирования;
- имеют более высокую стоимость сетей и сетевого оборудования.

Выбор архитектуры сети

Выбор архитектуры сети зависит от назначения сети, количества рабочих станций и от выполняемых на ней действий.

Следует выбрать одноранговую сеть, если:

- количество пользователей не превышает десяти;
- все машины находятся близко друг от друга;
- имеют место небольшие финансовые возможности;
- нет необходимости в специализированном сервере, таком как сервер БД, факс-сервер или какой-либо другой;
- нет возможности или необходимости в централизованном администрировании.

Следует выбрать клиент-серверную сеть, если:

- количество пользователей превышает десять;
- требуется централизованное управление, безопасность, управление ресурсами или резервное копирование;
- необходим специализированный сервер;
- нужен доступ к глобальной сети;
- требуется разделять ресурсы на уровне пользователей.

2.6 Лабораторная работа № 6 (4 часа)

Тема: «Архитектура Ethernet»

Цель работы: изучить базовую технологию локальных сетей – Ethernet. Приобрести навыки проектирования сложной вычислительной сети для организации, имеющей несколько филиалов, расположенных на достаточном расстоянии друг от друга.

Задачи работы:

1. Изучить технологию Ethernet;
2. Ознакомиться с методом доступа CSMA/CD;
3. Рассмотреть оптоволоконный Ethernet;
4. Ознакомиться с доменом коллизий.

Перечень приборов, материалов, используемых в лабораторной работе:
1. Компьютерная ЛВС.

Описание (ход) работы:

В 1980 г. Институт инженеров по электротехнике и радиоэлектронике (Institute of Electrical and Electronics Engineers — IEEE) организовал комитет 802, в состав которого вошли рабочие и исследовательские группы, занимающиеся стандартизацией локальных сетей. Результатом их работы стал перечень стандартов, регламентирующих проектирование локальных сетей:

• IEEE 802.1 — стандарты, относящиеся к управлению сетями;

• IEEE 802.2 — общий стандарт управления логическим соединением (Logical Link Control — LLC) и управления доступом к среде (Media Access Control — MAC);

• IEEE 802.3, 802.4, 802.5 — стандарты, определяющие управление доступом к среде передачи данных (Ethernet, FDDI, Token Ring);

• IEEE 802.6 – стандарт для городских сетей;

• IEEE 802.11 – стандарт беспроводных технологий передачи данных;

• IEEE 802.12 – стандарт, определяющий технологию передачи данных с методом доступа "приоритет запросов";

• и другие.

Технология Ethernet

Ethernet - это самый распространенный на сегодняшний день стандарт локальных сетей. В зависимости от типа физической среды стандарт IEEE 802.3 имеет различные модификации:

- 10Base-5;
- 10Base-2;
- 10Base-T;
- 10Base-FL;
- 10Base-FB;
- Fast Ethernet;
- Gigabit Ethernet.

Для передачи двоичной информации по кабелю для всех вариантов физического уровня технологии Ethernet, обеспечивающих пропускную способность 10 Мбит/с, используется манчестерский код.

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

Метод доступа CSMA/CD

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

Этот метод используется исключительно в сетях с общей шиной (к которым относятся и радиосети, породившие этот метод). Все компьютеры такой сети имеют непосредственный доступ к общей шине, поэтому она может быть использована для передачи данных между любыми двумя узлами сети. Простота схемы подключения - это один из факторов, определивших успех стандарта Ethernet. Говорят, что кабель, к

которому подключены все станции, работает в режиме *коллективного доступа (multiply-access, MA).*

Все данные, передаваемые по сети, помещаются в кадры определенной структуры и снабжаются уникальным адресом станции назначения. Затем кадр передается по кабелю. Все станции, подключенные к кабелю, могут распознать факт передачи кадра, и та станция, которая узнает собственный адрес в заголовках кадра, записывает его содержимое в свой внутренний буфер, обрабатывает полученные данные и посылает по кабелю кадр-ответ. Адрес станции-источника также включен в исходный кадр, поэтому станция-получатель знает, кому нужно послать ответ.

При описанном подходе возможна ситуация, когда две станции одновременно пытаются передать кадр данных по общему кабелю. Для уменьшения вероятности этой ситуации непосредственно перед отправкой кадра передающая станция слушает кабель (то есть принимает и анализирует возникающие на нем электрические сигналы), чтобы обнаружить, не передается ли уже по кабелю кадр данных от другой станции. Если *опознается несущая (carrier-sense, CS)*, то станция откладывает передачу своего кадра до окончания чужой передачи, и только потом пытается вновь его передать. Но даже при таком алгоритме две станции одновременно могут решить, что по шине в данный момент времени нет передачи, и начать одновременно передавать свои кадры. Говорят, что при этом происходит *коллизия*, так как содержимое обоих кадров сталкивается на общем кабеле, что приводит к искажению информации.

Чтобы корректно обработать коллизию, все станции одновременно наблюдают за возникающими на кабеле сигналами. Если передаваемые и наблюдаемые сигналы отличаются, то фиксируется *обнаружение коллизии (collision detection, CD)*. Для увеличения вероятности немедленного обнаружения коллизии всеми станциями сети, ситуация коллизии усиливается посылкой в сеть станциями, начавшими передачу своих кадров, специальной последовательности битов, называемой *jam-последовательностью*.

После обнаружения коллизии передающая станция обязана прекратить передачу и ожидать в течение случайного интервала времени, а затем может снова сделать попытку передачи кадра (рисунок 6.1).

Метод CSMA/CD определяет основные временные и логические соотношения, гарантирующие корректную работу всех станций в сети:

- Между двумя последовательно передаваемыми по общей шине кадрами информации должна выдерживаться пауза в 9.6 мкс; эта пауза нужна для приведения в исходное состояние сетевых адаптеров узлов, а также для предотвращения монопольного захвата среды передачи данных одной станцией.
- При обнаружении коллизии станция выдает в среду специальную 32-х битную последовательность (jam-последовательность), усиливающую явление коллизии для более надежного распознавания ее всеми узлами сети.
- После обнаружения коллизии каждый узел, который передавал кадр и столкнулся с коллизией, после некоторой задержки пытается повторно передать свой кадр. Узел делает максимально 16 попыток передачи этого кадра информации, после чего отказывается от его передачи. Величина задержки выбирается как равномерно распределенное случайное число из интервала, длина которого экспоненциально увеличивается с каждой попыткой. Такой алгоритм выбора величины задержки снижает вероятность коллизий и уменьшает интенсивность выдачи кадров в сеть при ее высокой загрузке.

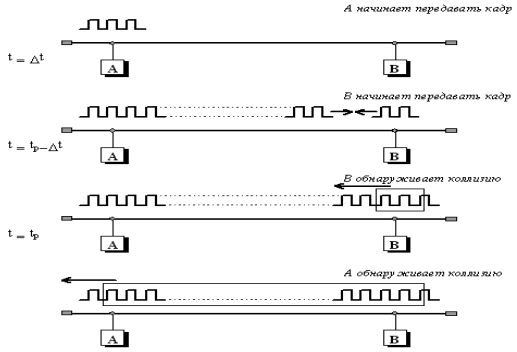


Рисунок 1 - Схема возникновения коллизии в методе случайного доступа CSMA/CD (tp - задержка распространения сигнала между станциями A и B)

Все параметры протокола Ethernet подобраны таким образом, чтобы при нормальной работе узлов сети коллизии всегда четко распознавались. Именно для этого минимальная длина поля данных кадра должна быть не менее 46 байт. Длина кабельной системы выбирается таким образом, чтобы за время передачи кадра минимальной длины сигнал коллизии успел бы распространиться до самого дальнего узла сети. Поэтому для скорости передачи данных 10 Мб/с, используемой в стандартах Ethernet, максимальное расстояние между двумя любыми узлами сети не должно превышать 2500 метров. Независимо от реализации физической среды, все сети Ethernet должны удовлетворять двум ограничениям, связанным с методом доступа: максимальное расстояние между двумя любыми узлами не должно превышать 2500 м; в сети не должно быть более 1024 узлов.

Оптоволоконный Ethernet

В качестве среды передачи данных 10 мегабитный Ethernet использует оптическое волокно. Оптоволоконные стандарты в качестве основного типа кабеля рекомендуют достаточно дешевое оптическое волокно, обладающее полосой пропускания 500-800 МГц при длине кабеля 1 км.

Функционально сеть Ethernet на оптическом кабеле состоит из тех же элементов, что и сеть стандарта 10Base-T - сетевых адаптеров, многопортового повторителя и отрезков кабеля, соединяющих адаптер с портом повторителя. Как и в случае витой пары, для соединения адаптера с повторителем *используются два* оптоволокна - одно соединяет выход T_x адаптера со входом R_x повторителя, а другое - вход R_x адаптера с выходом T_x повторителя.

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

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

Стандарт 10Base-FB предназначен только для соединения повторителей. Конечные узлы не могут использовать этот стандарт для присоединения к портам

концентратора. Между узлами сети можно установить до 5 повторителей 10Base-FB при максимальной длине одного сегмента 2000 м и максимальной длине сети 2740 м.

Как и в стандарте 10Base-T, оптоволоконные стандарты Ethernet разрешают соединять концентраторы только в древовидные иерархические структуры. Любые петли между портами концентраторов не допускаются.

Домен коллизий

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

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

Задание

Спроектировать вычислительную сеть для организации, имеющей несколько филиалов в различных городах. Используется 4 филиала в 4 населенных пунктах. Число компьютеров в каждом филиале - 5-5-8-6. Расстояние между сегментами - 25-20-25-20 км.

При проектировании использовать максимально возможное разнообразие коммуникационного оборудования. На схеме указать типы использованных сред передачи данных.

Определить несколько маршрутов продвижения пакета из одного узла в другой, составить таблицу маршрутизации.

Спроектировать сеть для организации, которая состоит из нескольких филиалов (варианты представлены в таблице А.1).

Таблица А.1

Вариант	Количество филиалов	Число компьютеров в каждом филиале	Количество населенных пунктов	Расстояние между сегментами, км
1	7	(12)-(24-5)	6	250-(0,4)
2	8	(10-10)-(24-11)	6	(1,8)-120-(0,15)
3	9	2-3-12-8-16	5	0,2-1,5-0,13-0,4
4	7	12-26-18	7	18-23
5	8	5-8-4-6	8	250-500-750
6	6	18-24	6	530
7	8	(10)-(8-7)-(29)	7	123-(0,2)-12
8	7	(8-19)-(23)	6	0,2-300
9	9	(12-12-14)-(8-12)	6	(0,3-0,4)-(0,25)
10	8	5-5-8-6	8	25-20-25-20
11	7	12-8-14	7	21-10
12	6	36-8	6	12-5
13	8	(12-2-3)-(15)	6	(1,8-0,4-0,9)-13
14	7	(10-8)-(16-14)	6	14-3

15	6	12-60	6	126
16	9	12-2-8-24-10	5	2-5-12-0,8
17	8	(11-5)-(3-12)	6	8-17
18	7	14-14-8	5	250-300
19	6	24-36	5	12
20	7	9-21-3	7	45-12-100

Необходимо добиться максимальной эффективности использования сети по критерию цена-качество-скорость.

2.7 Лабораторная работа №7 (2 часа)

Тема: **«**Многомашинные и многопроцессорные вычислительные системы**»**

2.7.1 Цель работы: изучить многомашинные и многопроцессорные вычислительные системы**.**

2.7.2 Задачи работы:

1. рассмотреть многомашинные и многопроцессорные вычислительные системы.

2.7.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. персональный компьютер, включенный в сеть IP, Microsoft Windows.

2.7.4 Описание (ход) работы:

В настоящее время тенденция в развитии микропроцессоров и систем, построенных на их основе, направлена на все большее повышение их производительности. Вычислительные возможности любой системы достигают своей наивысшей производительности благодаря двум факторам:

использованию высокоскоростных элементов и параллельному выполнению большого числа операций. Направления, связанные с повышением производительности отдельных микропроцессоров, мы рассматривали в предыдущих лекциях, а в этой лекции остановимся на вопросах распараллеливания обработки информации.

Существует несколько вариантов классификации систем параллельной обработки данных. По-видимому, самой ранней и наиболее известной является классификация архитектур вычислительных систем, предложенная в 1966 году М. Флинном. Классификация базируется на понятии потока, под которым понимается последовательность элементов, команд или данных, обрабатываемая процессором. На основе числа потоков команд и потоков данных выделяются четыре класса архитектур:

SISD, MISD, SIMD, MIMD.

SISD (sINgle INsTRuction sTReam / sINgle data sTReam) - одиночный поток команд и одиночный поток данных. К этому классу относятся прежде всего классические последовательные машины, или, иначе, машины фон-неймановского типа. В таких машинах есть только один поток команд, все команды обрабатываются последовательно друг за другом и каждая команда инициирует одну операцию с одним потоком данных. Не имеет значения тот факт, что для увеличения скорости обработки команд и скорости выполнения арифметических операций процессор может использовать конвейерную обработку. В таком понимании машины данного класса фактически не относятся к параллельным системам.

SIMD (sINgle INsTRuction sTReam / multIPle data sTReam) - одиночный поток команд и множественный поток данных. Применительно к одному микропроцессору этот подход реализован в MMX- и SSE- расширениях современных микропроцессоров.Микропроцессорные системы типа SIMD состоят из большого числа идентичных процессорных элементов, имеющих собственную память. Все процессорные элементы в такой машине выполняют одну и ту же программу. Это позволяет выполнять одну арифметическую операцию сразу над многими данными - элементамивектора.

Очевидно, что такая система, составленная из большого числа процессоров, может обеспечить существенное повышение производительности только на тех задачах, при решении которых все процессоры могут делать одну и ту же работу.

MISD (multIPe INsTRuction sTReam / sINgle data sTReam) - множественный поток команд и одиночный поток данных. Определениеподразумевает наличие в архитектуре многих процессоров, обрабатывающих один и тот же поток данных. Ряд исследователей к данному классу относят конвейерные машины.

MIMD (multIPe INsTRuction sTReam / multIPle data sTReam) - множественный поток команд и множественный поток данных. Базовоймоделью вычислений в этом случае является совокупность независимых процессов, эпизодически обращающихся к разделяемым данным. В такой системе каждый процессорный элемент выполняет свою программу достаточно независимо от других процессорных элементов.Архитектура MIMD дает большую гибкость: при наличии адекватной поддержки со стороны аппаратных средств и программного обеспечения MIMD может работать как однопользовательская система, обеспечивая высокопроизводительную обработку данных для одной прикладной задачи, как многопрограммная машина, выполняющая множество задач параллельно, и как некоторая комбинация этих возможностей. К тому же архитектура MIMD может использовать все преимущества современной микропроцессорной технологии на основе строгого учета соотношения стоимость/производительность. В действительности практически все современные многопроцессорные системы строятся на тех же микропроцессорах, которые можно найти в персональных компьютерах, рабочих станциях и небольших однопроцессорных серверах.

Как и любая другая, приведенная выше классификация несовершенна: существуют машины, прямо в нее не попадающие, имеются также важные признаки, которые в этой классификации не учтены. Рассмотрим классификацию многопроцессорных и многомашинных систем на основе другого признака - степени разделения вычислительных ресурсов системы.

В этом случае выделяют следующие 4 класса систем:

системы с симметричной мультипроцессорной обработкой (symmeTRic multIProcessINg), или SMP-системы;

системы, построенные по технологии неоднородного доступа к памяти (non-un IForm memory access), или NUMA-системы;

кластеры;

системы вычислений с массовым параллелизмом (massively parallel processor), или MPP-системы.

Самым высоким уровнем интеграции ресурсов обладает система с симметричной мультипроцессорной обработкой, или SMP-система (рис. 13.1).

В этой архитектуре все процессоры имеют равноправный доступ ко всему пространству оперативной памяти и ввода/вывода. Поэтому SMP-архитектура называется симметричной. Ее интерфейсы доступа к пространству ввода/вывода и ОП, система управления кэш-памятью,системное ПО и т. п. построены таким образом, чтобы обеспечить согласованный доступ к разделяемым ресурсам. Соответствующиемеханизмы блокировки заложены и в шинном интерфейсе, и в компонентах операционной системы, и при построении кэша.

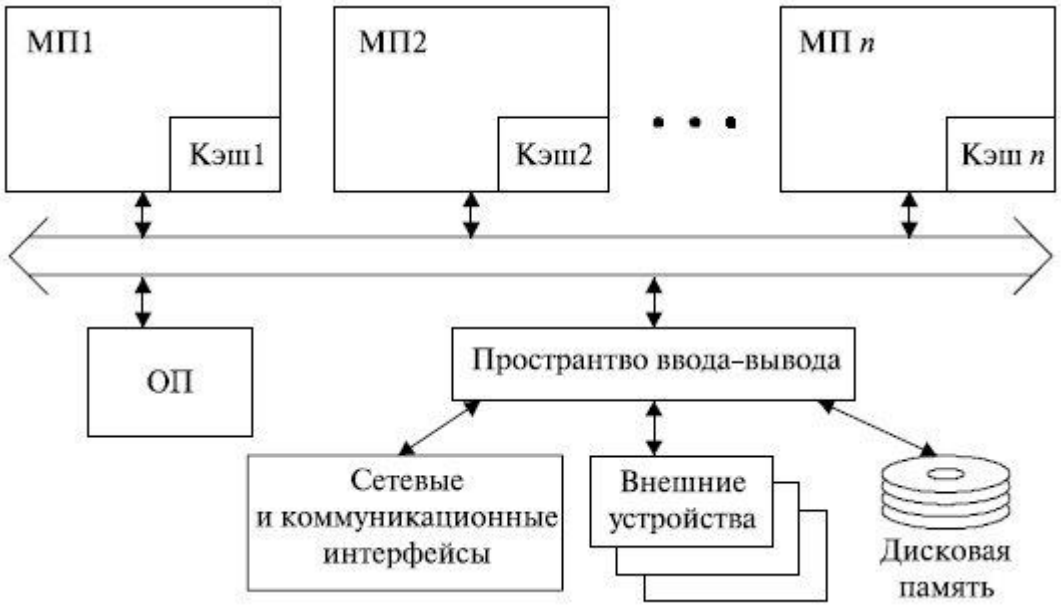


Рис. 13.1. Система с симметричной мультипроцессорной обработкой

С точки зрения прикладной задачи, SMP-система представляет собой единый вычислительный комплекс с вычислительными ресурсами, пропорциональными количеству процессоров. Распараллеливание вычислений обеспечивается операционной системой, установленной на одном из процессоров. Вся система работает под управлением единой ОС (обычно UNIX-подобной, но для Intel-платформ поддерживается WINdows NT).

ОС автоматически в процессе работы распределяет процессы по процессорным ядрам, оптимизируя использование ресурсов. Ядра задействуются равномерно, и прикладные программы могут выполняться параллельно на всем множестве ядер. При этом достигается максимальное быстродействие системы. Важно, что для синхронизации приложений вместо сложных механизмов и протоколовмежпроцессорной коммуникации применяются стандартные функции ОС. Таким образом, проще реализовать проекты с распараллеливанием программных потоков. Общая для совокупности ядер ОС позволяет с помощью служебных инструментов собирать статистику, единую для всей архитектуры. Соответственно, можно облегчить отладку и оптимизацию приложений на этапе разработки или масштабирования для других форм многопроцессорной обработки.

В общем случае приложение, написанное для однопроцессорной системы, не требует модификации при его переносе в мультипроцессорную среду. Однако для оптимальной работы программы или частей ОС они переписываются специально для работы в мультипроцессорной среде.

Сравнительно небольшое количество процессоров в таких машинах позволяет иметь одну централизованную общую память и объединить процессоры и память с помощью одной шины.

Сдерживающим фактором в подобных системах является пропускная способность магистрали, что приводит к их плохой масштабируемости. Причиной этого является то, что в каждый момент времени шина способна обрабатывать только одну транзакцию, вследствие чего возникают проблемы разрешения конфликтов при одновременном обращении нескольких процессоров к одним и тем же областям общей физической памяти. Вычислительные элементы начинают мешать друг другу. Когда произойдет такой конфликт, зависит от скорости связи и от количества вычислительных элементов. Кроме того, системная шина имеет ограниченное число слотов. Все это очевидно препятствуетувеличению производительности при увеличении числа процессоров. В реальных системах можно задействовать не более 32 процессоров.

В современных микропроцессорах поддержка построения мультипроцессорной системы закладывается на уровне аппаратной реализации МП, что делает многопроцессорные системы сравнительно недорогими.

Так, для обеспечения возможности работы на общую магистраль каждый микропроцессор фирмы Intel начиная с Pentium Pro имеет встроенную поддержку двухразрядного идентификатора процессора - APIC (Advanced Programmable INTerrupt ConTRoller). По умолчанию CPUс самым высоким номером идентификатора становится процессором начальной загрузки. Такая идентификация облегчает арбитраж шины данных в SMP-системе. Подобные средства мы видели и в МП Power4, где на аппаратном уровне поддерживается создание микросхемного модуля MCM из 4 микропроцессоров, включающего в совокупности 8 процессорных ядер.

Сегодня SMP широко применяют в многопроцессорных суперкомпьютерах и серверных приложениях. Однако если необходимо детерминированное исполнение программ в реальном масштабе времени, например, при визуализации мультимедийных данных, возможности сугубосимметричной обработки весьма ограничены. Может возникнуть ситуация, когда приложения, выполняемые на различных ядрах, обращаются к одному ресурсу ОС. В этом случае доступ получит только одно из ядер.

Остальные будут простаивать до высвобождения критической области.

Естественно, при этом резко снижается производительность приложений реального времени.

Исчерпание производительности системной шины в SMP-системах при доступе большого числа процессоров к общему пространству оперативной памяти и принципиальные ограничения шинной технологии стали причиной сдерживания роста производительности SMP-систем. На данный момент эта проблема получила два решения. Первое - замена системной шины на высокопроизводительный коммутатор, обеспечивающий одновременный неблокирующий доступ к различным участкам памяти. Второе решение предлагает технология NUMA.

Система, построенная по технологии NUMA, представляет собой набор узлов, каждый из которых, по сути, является функционально законченным однопроцессорным или SMP-компьютером. Каждый имеет свое локальное пространство оперативной памяти и ввода/вывода. Но с помощью специальной логики каждый имеет доступ к пространству оперативной памяти и ввода/вывода любого другого узла (рис. 13.2). Физически отдельные устройства памяти могут адресоваться как логически единое адресное пространство - это означает, что любойпроцессор может выполнять обращения к любым ячейкам памяти, в предположении, что он имеет соответствующие права доступа. Поэтому иногда такие системы называются системами с распределенной разделяемой памятью (DSM - disTRibuted shared memory).

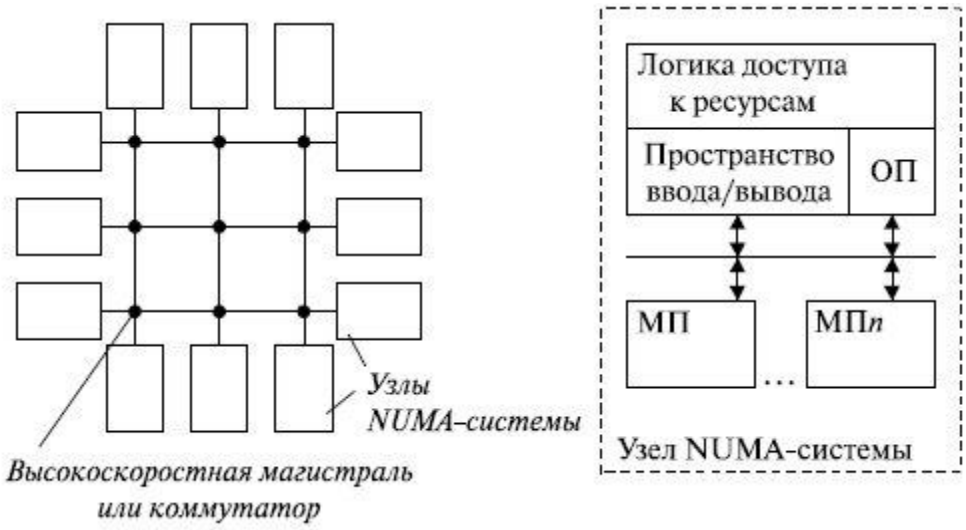


Рис. 13.2. Система, построенная по технологии неоднородного до ступа к памяти

При такой организации память каждого узла системы имеет свою адресацию в адресном пространстве всей системы. Логика доступа к ресурсам определяет, к памяти какого узла относится выработанный процессором адрес. Если он не принадлежит памяти данного узла, организуется обращение к другому узлу согласно заложенной в логике доступа карте адресов. При этом доступ к локальной памяти осуществляется в несколько раз быстрее, чем к удаленной.

При использовании наиболее распространенного сейчас варианта cc-NUMA (cache-coherent NUMA - неоднородный доступ к памяти с согласованием содержимого кэш-памяти) обеспечивается кэширование данных оперативной памяти других узлов.

Обычно вся система работает под управлением единой ОС, как в SMP. Возможны также варианты динамического разделения системы, когда отдельные разделы системы работают под управлением разных ОС.

Довольно большое время доступа к оперативной памяти соседних узлов по сравнению с доступом к ОП своего узла в NUMA-системах на настоящий момент делает такое использование не вполне оптимальным.

Так что полной функциональностью SMP-систем NUMA-компьютеры на сегодняшний день не обладают. Однако среди систем общего назначения NUMA-системы имеют один из наиболее высоких показателей по масштабируемости и, соответственно, попроизводительности. На сегодня максимальное число процессоров в cc-NUMA-системах может превышать 1000 (серия OrigIN3000). Один из наиболее производительных суперкомпьютеров - Tera 10 - имеет производительностью 60 Тфлопс и состоит из 544 SMP-узлов, в каждом из которых находится от 8 до 16 процессоров Itanium 2.

Следующим уровнем в иерархии параллельных систем являются комплексы, также состоящие из отдельных машин, но лишь частично разделяющие некоторые ресурсы. Речь идет о кластерах.

2.8 Лабораторная работа № 8 (2 часа)

Тема: «Технология Token Ring»

Цель работы: изучить базовую технологию локальных сетей Token Ring, маркерный метод доступа к разделяемой среде.

Задачи работы:

1. Изучить технологию Token Ring;
2. Ознакомиться с маркерным методом доступа к разделяемой среде;
3. Рассмотреть физический уровень технологии Token Ring.

Перечень приборов, материалов, используемых в лабораторной работе:
1. Компьютерная ЛВС.
Описание (ход) работы:

Сети Token Ring, так же как и сети Ethernet, характеризует разделяемая среда передачи данных, которая в данном случае состоит из отрезков кабеля, соединяющих все станции сети в кольцо. Кольцо рассматривается как общий разделяемый ресурс, и для доступа к нему требуется не случайный алгоритм, как в сетях Ethernet, а детерминированный, основанный на передаче станциям права на использование кольца в определенном порядке. Это право передается с помощью кадра специального формата, называемого *маркером* или *токеном (token)*.

Технология Token Ring был разработана компанией IBM в 1984 году, а затем передана в качестве проекта стандарта в комитет IEEE 802, который на ее основе принял в 1985 году стандарт 802.5.

Сети Token Ring работают с двумя битовыми скоростями - 4 и 16 Мбит/с. Смешение станций, работающих на различных скоростях, в одном кольце не допускается.

Технология Token Ring является более сложной технологией, чем Ethernet. Она обладает свойствами отказоустойчивости. В сети Token Ring определены процедуры контроля работы сети, которые используют обратную связь кольцеобразной структуры - посланный кадр всегда возвращается в станцию - отправитель. В некоторых случаях обнаруженные ошибки в работе сети устраняются автоматически, например может быть восстановлен потерянный маркер. В других случаях ошибки только фиксируются, а их устранение выполняется вручную обслуживающим персоналом.

Для контроля сети одна из станций выполняет роль так называемого *активного монитора*. Активный монитор выбирается во время инициализации кольца как станция с максимальным значением MAC-адреса, Если активный монитор выходит из строя, процедура инициализации кольца повторяется и выбирается новый активный монитор. Чтобы сеть могла обнаружить отказ активного монитора, последний в работоспособном состоянии каждые 3 секунды генерирует специальный кадр своего присутствия. Если этот кадр не появляется в сети более 7 секунд, то остальные станции сети начинают процедуру выборов нового активного монитора.

Маркерный метод доступа к разделяемой среде

В сетях с *маркерным методом доступа* (а к ним, кроме сетей Token Ring, относятся сети FDDI) право на доступ к среде передается циклически от станции к станции по логическому кольцу.

В сети Token Ring кольцо образуется отрезками кабеля, соединяющими соседние станции. Таким образом, каждая станция связана со своей предшествующей и последующей станцией и может непосредственно обмениваться данными только с ними. Для обеспечения доступа станций к физической среде по кольцу циркулирует кадр специального формата и назначения - маркер. В сети Token Ring любая станция всегда непосредственно получает данные только от одной станции - той, которая является предыдущей в кольце. Такая станция называется *ближайшим активным соседом, расположенным выше по потоку* (данных) - *Nearest Active Upstream Neighbor, NAUN*. Передачу же данных станция всегда осуществляет своему ближайшему соседу вниз по потоку данных.

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

проходят по кольцу всегда в одном направлении от одной станции к другой. Кадр снабжен адресом назначения и адресом источника.

Все станции кольца ретранслируют кадр побитно, как повторители. Если кадр проходит через станцию назначения, то, распознав свой адрес, эта станция копирует кадр в свой внутренний буфер и вставляет в кадр признак подтверждения приема. Станция, выдавшая кадр данных в кольцо, при обратном его получении с подтверждением приема изымает этот кадр из кольца и передает в сеть новый маркер для обеспечения возможности другим станциям сети передавать данные. Такой алгоритм доступа применяется в сетях Token Ring со скоростью работы 4 Мбит/с, описанных в стандарте 802.5.

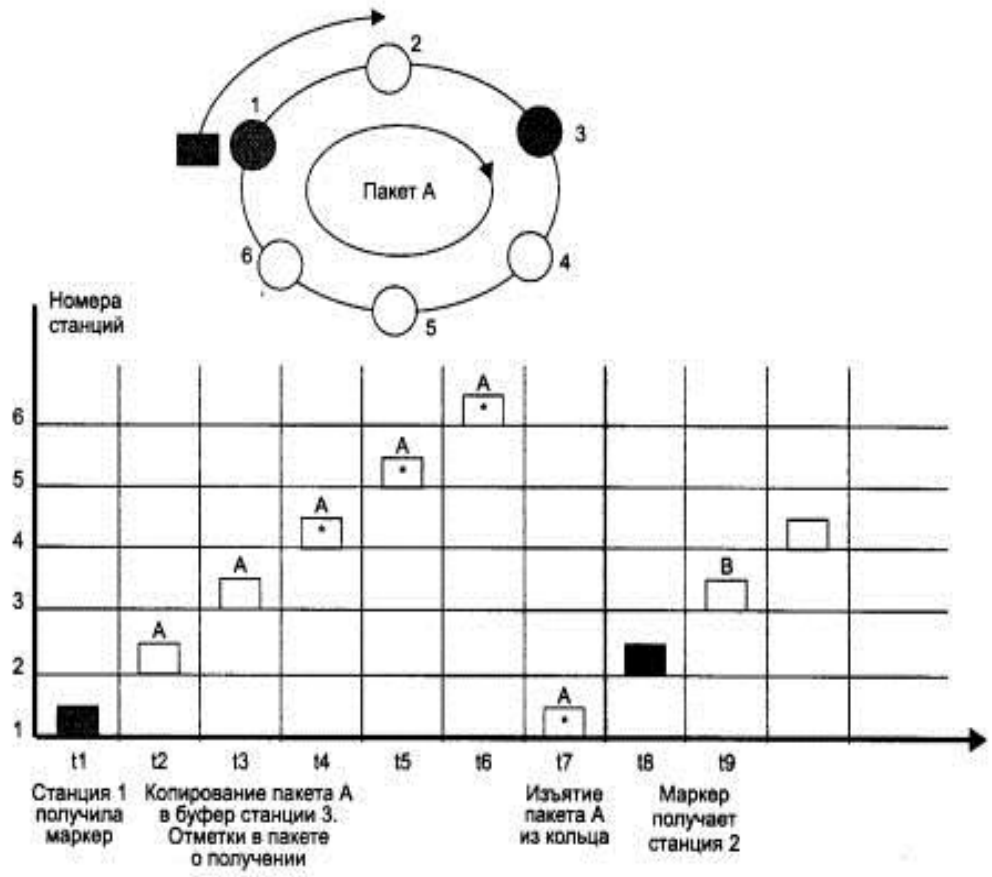


Рисунок 1 – Принцип маркерного доступа

На рисунке 1 описанный алгоритм доступа к среде иллюстрируется временной диаграммой. Здесь показана передача пакета А в кольце, состоящем из 6 станций, от станции *1* к станции *3*. После прохождения станции назначения *3* в пакете А устанавливаются два признака - признак распознавания адреса и признак копирования пакета в буфер (что на рисунке отмечено звездочкой внутри пакета). После возвращения пакета в станцию *1* отправитель распознает свой пакет по адресу источника и удаляет пакет из кольца. Установленные станцией *3* признаки говорят станции-отправителю о том, что пакет дошел до адресата и был успешно скопирован им в свой буфер.

Время владения разделяемой средой в сети Token Ring ограничивается *временем удержания маркера*, после истечения которого станция обязана прекратить передачу собственных данных (текущий кадр разрешается завершить) и передать маркер далее по кольцу. Станция может успеть передать за время удержания маркера один или несколько кадров в зависимости от размера кадров и величины времени удержания маркера. Обычно время удержания маркера по умолчанию равно 10 мс.

Для различных видов сообщений передаваемым кадрам могут назначаться различные *приоритеты*: от 0 (низший) до 7 (высший). Решение о приоритете конкретного кадра принимает передающая станция. Маркер также всегда имеет некоторый уровень текущего приоритета. Станция имеет право захватить переданный ей маркер только в том случае, если приоритет кадра, который она хочет передать, выше (или равен) приоритета маркера. В противном случае станция обязана передать маркер следующей по кольцу станции.

За наличие в сети маркера, причем единственной его копии, отвечает активный монитор. Если активный монитор не получает маркер в течение длительного времени (например, 2,6 с), то он порождает новый маркер.

Физический уровень технологии Token Ring

Стандарт Token Ring фирмы IBM изначально предусматривал построение связей в сети с помощью концентраторов, называемых MAU (Multistation Access Unit) или MSAU (Multi-Station Access Unit), то есть устройствами многостанционного доступа (рисунок 6.3). Сеть Token Ring может включать до 260 узлов.

Концентратор Token Ring может быть активным или пассивным. Пассивный концентратор просто соединяет порты внутренними связями так, чтобы станции, подключаемые к этим портам, образовали кольцо. Такое устройство можно считать простым кроссовым блоком за одним исключением - MSAU обеспечивает обход какого-либо порта, когда присоединенный к этому порту компьютер выключают. Такая функция необходима для обеспечения связности кольца вне зависимости от состояния подключенных компьютеров.

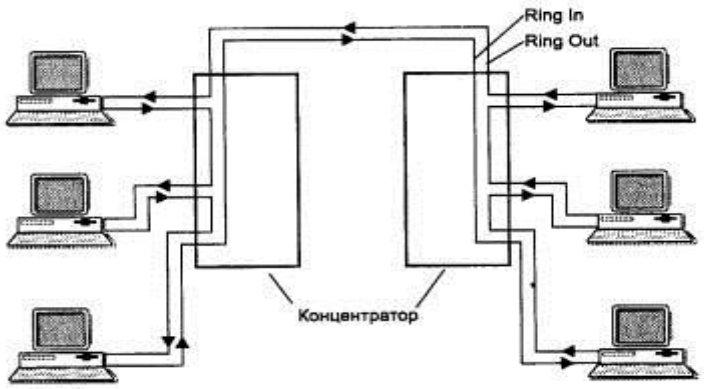


Рисунок 2 – Физическая конфигурация сети Token Ring

Активный концентратор выполняет функции регенерации сигналов и поэтому иногда называется повторителем, как в стандарте Ethernet.

Возникает вопрос – если концентратор является пассивным устройством, то каким образом обеспечивается качественная передача сигналов на большие расстояния, которые возникают при включении в сеть нескольких сот компьютеров? Ответ состоит в том, что роль усилителя сигналов в этом случае берет на себя каждый сетевой адаптер, а роль ресинхронизирующего блока выполняет сетевой адаптер активного монитора кольца. Каждый сетевой адаптер Token Ring имеет блок повторения, который умеет регенерировать и ресинхронизировать сигналы, однако последнюю функцию выполняет в кольце только блок повторения активного монитора.

Максимальная длина кольца Token Ring составляет 4000 м. Ограничения на максимальную длину кольца и количество станций в кольце в технологии Token Ring не являются такими жесткими, как в технологии Ethernet. Здесь эти ограничения во многом

связаны со временем оборота маркера по кольцу (но не только – есть и другие соображения, диктующие выбор ограничений). Так, если кольцо состоит из 260 станций, то при времени удержания маркера в 10 мс маркер вернется в активный монитор в худшем случае через 2,6 с, а это время как раз составляет тайм-аут контроля оборота маркера.

Контрольные вопросы

1) Что подразумевается под термином “Token Ring”.
2) Объясните технологию Token Ring.
3) Сравните технологии Ethernet и Token Ring.

2.9 Лабораторная работа № 9 (2 часа)

Тема: «Архитектура FDDI»

Цель работы: изучить базовые технологии локальных сетей FDDI.

Задачи работы:
1. Изучить технологию FDDI;
2. Ознакомиться с особенностями метода доступа FDDI.

Перечень приборов, материалов, используемых в лабораторной работе:
1. Компьютерная ЛВС.

Описание (ход) работы:

Технология FDDI

Технология *FDDI (Fiber Distributed Data Interface)*- оптоволоконный интерфейс распределенных данных - это первая технология локальных сетей, в которой средой передачи данных является волоконно-оптический кабель.

Технология FDDI во многом основывается на технологии Token Ring, развивая и совершенствуя ее основные идеи.

Сеть FDDI строится на основе двух оптоволоконных колец, которые образуют основной и резервный пути передачи данных между узлами сети. Наличие двух колец - это основной способ повышения отказоустойчивости в сети FDDI, и узлы, которые хотят воспользоваться этим повышенным потенциалом надежности, должны быть подключены к обоим кольцам.

В нормальном режиме работы сети данные проходят через все узлы и все участки кабеля только первичного (Primary) кольца, этот режим назван режимом *Thru* - «сквозным» или «транзитным». Вторичное кольцо (Secondary) в этом режиме не используется.

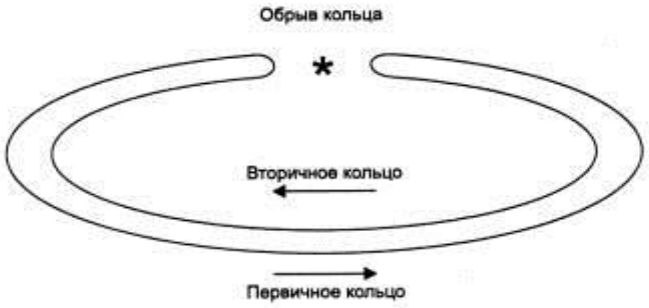


Рисунок 1 - Реконфигурация колец FDDI при отказе

В случае какого-либо вида отказа, когда часть первичного кольца не может передавать данные (например, обрыв кабеля или отказ узла), первичное кольцо объединяется со вторичным (рисунок 1), вновь образуя единое кольцо. Этот режим работы сети называется *Wrap,* то есть «свертывание» или «сворачивание» колец. Операция свертывания производится средствами концентраторов и/или сетевых адаптеров FDDI. Для упрощения этой процедуры данные по первичному кольцу всегда передаются в одном направлении (на диаграммах это направление изображается против часовой стрелки), а по вторичному - в обратном (изображается по часовой стрелке). Поэтому при образовании общего кольца из двух колец передатчики станций по-прежнему остаются подключенными к приемникам соседних станций, что позволяет правильно передавать и принимать информацию соседними станциями.

Технология FDDI дополняет механизмы обнаружения отказов технологии Token Ring механизмами реконфигурации пути передачи данных в сети, основанными на наличии резервных связей, обеспечиваемых вторым кольцом.

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

Отличия метода доступа заключаются в том, что время удержания маркера в сети FDDI не является постоянной величиной, как в сети Token Ring. Это время зависит от загрузки кольца - при небольшой загрузке оно увеличивается, а при больших перегрузках может уменьшаться до нуля. Эти изменения в методе доступа касаются только асинхронного трафика, который не критичен к небольшим задержкам передачи кадров. Для синхронного трафика время удержания маркера по-прежнему остается фиксированной величиной. Механизм приоритетов кадров, аналогичный принятому в технологии Token Ring, в технологии FDDI отсутствует. Разработчики технологии решили, что деление трафика на 8 уровней приоритетов избыточно и достаточно разделить трафик на два класса - асинхронный и синхронный, последний из которых обслуживается всегда, даже при перегрузках кольца.

В остальном пересылка кадров между станциями кольца на уровне MAC полностью соответствует технологии Token Ring. Станции FDDI применяют алгоритм раннего освобождения маркера, как и сети Token Ring со скоростью 16 Мбит/с.

Скорость передачи данных для сетей FDDI составляет 100 Мбит/с. Максимальное число узлов составляет 500. При использовании в качестве физической среды передачи данных многомодового оптоволоконного кабеля расстояние между узлами сети может доходить до 2 км, а при использовании одномодового кабеля – до 40 км. В случае кабеля на основе витых пар пятой категории это расстояние не превышает 100 м. максимальный диаметр двойного кольца не должен превышать 100 км.

Адреса уровня MAC имеют стандартный для технологий IEEE 802 формат. Формат кадра FDDI близок к формату кадра Token Ring, основные отличия заключаются в отсутствии полей приоритетов.

Особенности метода доступа FDDI.

Для передачи синхронных кадров станция всегда имеет право захватить маркер при его поступлении. При этом время удержания маркера имеет заранее заданную фиксированную величину.

Если же станции кольца FDDI нужно передать асинхронный кадр (тип кадра определяется протоколами верхних уровней), то для выяснения возможности захвата маркера при его очередном поступлении станция должна измерить интервал времени, который прошел с момента предыдущего прихода маркера. Этот интервал называется временем оборота маркера. Если в технологии Token Ring максимально допустимое время оборота маркера является фиксированной величиной (2,6 с из расчета 260 станций в кольце), то в технологии FDDI станции договариваются о его величине во время инициализации кольца. Каждая станция может предложить свое значение, в результате для кольца устанавливается минимальное из предложенных станциями времен. Это позволяет учитывать потребности приложений, работающих на станциях. Обычно синхронным приложениям (приложениям реального времени) нужно чаще передавать данные в сеть небольшими порциями, а асинхронным приложениям лучше получать доступ к сети реже, но большими порциями. Предпочтение отдается станциям, передающим синхронный трафик.

Для обеспечения отказоустойчивости в стандарте FDDI предусмотрено создание двух оптоволоконных колец - первичного и вторичного. В стандарте FDDI допускаются два вида подсоединения станций к сети. Одновременное подключение к первичному и вторичному кольцам называется двойным подключением - Dual Attachment, DA. Подключение только к первичному кольцу называется одиночным подключением - Single Attachment, SA.

В стандарте FDDI предусмотрено наличие в сети конечных узлов – станций и также концентраторов. Для станций и концентраторов допустим любой вид подключения к сети - как одиночный, так и двойной. Обычно концентраторы имеют двойное подключение, а станции - одинарное, как это показано на рисунке 2, хотя это и не обязательно. Чтобы устройства легче было правильно присоединять к сети, их разъемы маркируются. Разъемы типа A и B должны быть у устройств с двойным подключением, разъем M (Master) имеется у концентратора для одиночного подключения станции, у которой ответный разъем должен иметь тип S (Slave).

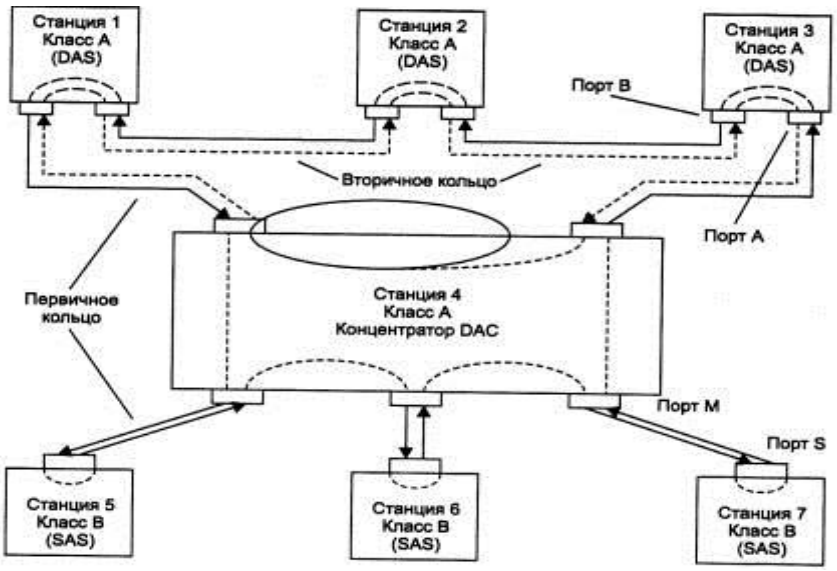


Рисунок 2 – Подключение узлов к кольцам FDDI. SAS (Single Attachment Station), DAS (Dual Attachment Station), SAC (Single Attachment Concentrator) и DAC (Dual Attachment Concentrator).

Стандарт **FDDI** (*Fiber Distributed Data Interface*□ волоконно-оптический интерфейс передачи данных), разработанный в середине 80-х годов комитетом X3T9.5 ANSI, определяет кольцевую сеть с маркерным доступом и скоростью передачи до 100 Мбит/с на основе волоконно-оптического кабеля, способною охватить очень большую площадь (до 100 км).

Стандарт FDDI во многом основывается на технологии Token Ring (стандарт IEEE 802.5) и обеспечивает совместимость с ней, т.к. у обеих технологий одинаковые форматы кадров. Однако у этих технологий имеются существенные различия.

Стек FDDI определяет физический уровень и подуровень доступа к среде передачи (MAC). Физический уровень разбит на протокол физического уровня (Physical Layer Protocol, PHY), который отвечает за работу схем кодирования данных, и на подуровень физического уровня, зависящий от среды передачи (Physical Medium Dependent, PMD), на котором реализованы спецификации передачи. Особенностью стека FDDI является наличие уровня управления станциями (Station Management, SMT). Он отвечает за удаление и подключение рабочих станций, обнаружение и устранение неисправностей, сбор статистической информации о работе сети.

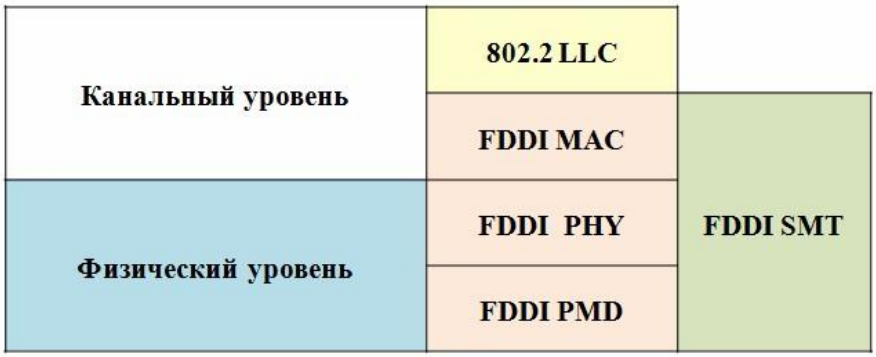


Рисунок 3. Стек FDDI

Сети FDDI характеризуются встроенной избыточностью, что обеспечивает их высокую отказоустойчивость. Сеть FDDI строится на основе двух колец, которые образуют основной и резервный пути передачи данных между узлами сети. Данные в кольцах циркулируют в разных направлениях. Одно кольцо считается основным (первичным). По нему данные передаются при нормальной работе. Второе кольцо (вторичное) вспомогательное, по нему данные передаются в случае обрыва в первом кольце. В случае какого-либо вида отказа, когда часть первого кольца не может передавать данные (например, обрыв кабеля или отказ узла), сеть выполняет «свертывание» колец объединяет первое кольцо со вторым, образуя единое кольцо.

Основными компонентами сети FDDI являются станции и концентраторы. Для подключения станций и концентраторов к сети может быть использован один из двух способов:

- **Одиночное подключение** (Single Attachment, SA) подключение только к первичному кольцу. Станция и концентратор, подключенные данным способом, называются соответственно станцией одиночного подключения (Single Attachment Station, SAS) и концентратором одиночного подключения (Single Attachment Concentrator, SAC).
- **Двойное подключение** (Dual Attachment, DA) одновременное подключение к первичному и вторичному кольцам. Станция и концентратор, подключенные таким способом, называются соответственно станцией двойного подключения (Dual Attachment Station, DAS) и концентратором двойного подключения (Dual Attachment Concentrator, DAC).

В качестве среды передачи в сетях FDDI используется одномодовый и многомодовый волоконно-оптический кабель. Максимальное количество станций в кольце – 500. Максимальное расстояние между узлами может составлять 2 км при использовании многомодового кабеля и 20 км – при использовании одномодового. Максимальная протяженность сети – 100 км.

К преимуществам технологии FDDI можно отнести высокую отказоустойчивость. К недостаткам – двойной расход кабеля.

В настоящее время эта технология считается устаревшей.

3. Методические указания по проведению практических занятий

3.1 Практическое занятие № 1 (2 часа)

Тема: «Общие сведения о компьютерных сетях»

3.1.1 Задачи работы:

1. Ознакомится с персональными, локальными, глобальными и городскими вычислительными сетями.;

2. Изучить топологии сетей и их достоинства и не достоинства.

3.1.2 Краткое описание практического занятия:

Ознакомление с персональными, локальными, глобальными и городскими вычислительными сетями.;

Изучение топологии сетей и их достоинства и не достоинства.

3.1.3 Результаты и выводы:

Классификация сетей ЭВМ (компьютерных сетей), как любых больших и сложных систем, может быть выполнена на основе различных признаков, в качестве которых могут быть использованы (рис.1):

- размер (территориальный охват) сети;
- принадлежность;
- назначение;
- область применения.

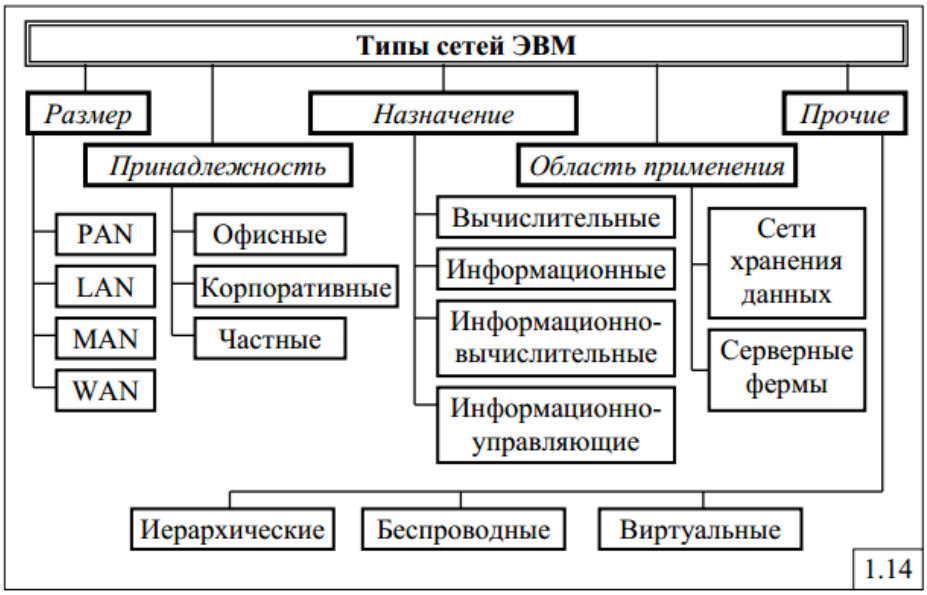


Рисунок 1.Типы сетей ЭВМ

1. **По размеру (территориальному охвату)** сети ЭВМ делятся на:

- персональные;
- локальные;
- городские (региональные).
- глобальные.

Персональная сеть (PersonalAreaNetwork, PAN) — это сеть,объединяющая персональные электронные устройства пользователя (телефоны, карманные персональные компьютеры, смартфоны, ноутбуки и т.п.) и характеризующаяся:

- небольшим числом абонентов;
- малым радиусом действия (до нескольких десятков метров);

- некритичностью к отказам.

К стандартам таких сетей в настоящее время относятся Bluetooth, Zigbee, Пиконет.

Локальная вычислительная сеть (ЛВС) (LocalAreaNetwork, LAN)– сеть со скоростью передачи данных , как правило, не менее 1 Мбит/с, обеспечивающая связь на небольших расстояниях – от нескольких десятков метров до нескольких километров. Оборудование, подключаемое к ЛВС, может находиться в одном или нескольких соседних зданиях.

Примеры ЛВС: Ethernet, Token Ring.

Городская вычислительная сеть (MetropolitanAreaNetwork, MAN) – сеть, промежуточная по размеру между ЛВС и глобальной сетью.

Протоколы и кабельная система для городской вычислительной сети описываются в стандартах комитета IEEE 802.6. MAN реализуется на основе протокола DQDB (DistributedQueueDualBus) – двойная шина сраспределенной очередью и использует волоконно-оптический кабель для передачи данных со скоростью 100 Мбит/с на территории до 100 км2. MAN может применяться для объединения в одну сеть группы сетей, расположенных в разных зданиях. Последние разработки, связанные с высокоскоростным беспроводным доступом в соответствии со стандартом IEEE 802.16, привели к созданию MAN в виде широкополосных беспроводных ЛВС.

Глобальная сеть (WideAreaNetwork, WAN) – в отличие от ЛВС охватывает большую территорию и представляет собой объединение нескольких ЛВС, связанных с помощью специального сетевого оборудования (маршрутизаторов, коммутаторов и шлюзов), образующих в случае использования высокоскоростных каналов магистральную сеть передачи данных (магистральную сеть связи). Наиболее широкое применение находят глобальные сети для нужд информационного обмена в коммерческих, научных и других профессиональных целях.

Для построения глобальных сетей могут использоваться различные сетевые технологии, в том числе TCP/IP, X.25, FrameRelay, ATM, MPLS.

Настоящей глобальной сетью, пожалуй, можно считать только сеть Интернет. Вряд ли глобальной можно считать сеть, объединяющую 2-3 ЛВС, находящиеся в разных городах, расположенных на расстоянии нескольких десятков или даже сотен километров друг от друга. Однако, поскольку для построения такой «простой» сети используются обычно те же сетевые технологии и технические средства, что и в сети Интернет, то такие сети обычно тоже относят к классу глобальных сетей.

2. **По принадлежности** сети ЭВМ делятся на:

- офисные – сети, расположенные на территории офиса компании, ограниченной обычно пределами одного здания, и построенные на технологиях LAN;
- корпоративные (ведомственные) – сети, представляющие собой объединение нескольких офисных сетей компании, расположенных в разных территориально разнесенных зданиях, находящихся возможно в разных городах и регионах, и построенные на технологиях MAN или WAN;
- частные – сети, построенные обычно на технологии виртуальной частной сети (VirtualPrivateNetwork, VPN), позволяющей обеспечить одно или несколько сетевых соединений, которые могут быть трёх видов: узел-узел, узел-сеть и сеть-сеть, образующих логическую сеть поверхдругой сети (например, Интернет).

3. **По назначению** сети ЭВМ делятся на:

- вычислительные, предназначенные для решения задач пользователей, ориентированных, в основном, на вычисления;
- информационные, ориентированные на предоставление информационных услуг; примерами таких сетей могут служить сети, предоставляющие справочные и библиотечные услуги;

2. Топология сетей.

Термин ***топология сети*** означает способ соединения компьютеров в сеть. Вы также можете услышать другие названия – ***структура сети*** или***конфигурация сети*** (это одно и то же). Кроме того, понятие топологии включает множество правил, которые определяют места размещения компьютеров, способы прокладки кабеля, способы размещения связующего оборудования и многое другое. На сегодняшний день сформировались и устоялись несколько основных топологий. Из них можно отметить "**шину**", "**кольцо**" и "**звезду**".

Топология "шина"

Топология ***шина*** (рис.2) (или, как ее еще часто называют ***общая шина*** или ***магистраль***) предполагает использование одного кабеля, к которому подсоединены все рабочие станции.

Рисунок 2. Топология шина.

Общий кабель используется всеми станциями по очереди. Все сообщения, посылаемые отдельными рабочими станциями, принимаются и прослушиваются всеми остальными компьютерами, подключенными к сети. Из этого потока каждая рабочая станция отбирает адресованные только ей сообщения.

Достоинства топологии "шина":

- простота настройки;
- относительная простота монтажа и дешевизна, если все рабочие станции расположены рядом;
- выход из строя одной или нескольких рабочих станций никак не отражается на работе всей сети.

Недостатки топологии "шина":

- неполадки шины в любом месте (обрыв кабеля, выход из строя сетевого коннектора) приводят к неработоспособности сети;
- сложность поиска неисправностей;
- низкая производительность – в каждый момент времени только один компьютер может передавать данные в сеть, с увеличением числа рабочих станций производительность сети падает;
- плохая масштабируемость – для добавления новых рабочих станций необходимо заменять участки существующей шины.

Именно по топологии "шина" строились локальные сети на коаксиальном кабеле. В этом случае в качестве шины выступали отрезки коаксиального кабеля, соединенные Т-коннекторами. Шина прокладывалась через все помещения и подходила к каждому компьютеру. Боковой вывод Т-коннектора вставлялся в разъем на сетевой карте. Сейчас такие сети безнадежно устарели и повсюду заменены "звездой" на витой паре, однако оборудование под коаксиальный кабель еще можно увидеть на некоторых предприятиях.

Топология "кольцо"

Кольцо – это топология локальной сети, в которой рабочие станции подключены последовательно друг к другу, образуя замкнутое кольцо. Данные передаются от одной

рабочей станции к другой в одном направлении (по кругу) (рис.3). Каждый ПК работает как повторитель, ретранслируя сообщения к следующему ПК, т.е. данные передаются от одного компьютера к другому как бы по эстафете.

Рисунок 3. Топология кольцо.

Если компьютер получает данные, предназначенные для другого компьютера – он передает их дальше по кольцу, в ином случае они дальше не передаются.

Достоинства кольцевой топологии:

- простота установки;
- практически полное отсутствие дополнительного оборудования;
- возможность устойчивой работы без существенного падения скорости передачи данных при интенсивной загрузке сети.

Однако “кольцо” имеет и существенные недостатки:

- каждая рабочая станция должна активно участвовать в пересылке информации; в случае выхода из строя хотя бы одной из них или обрыва кабеля – работа всей сети останавливается;
- подключение новой рабочей станции требует краткосрочного выключения сети, поскольку во время установки нового ПК кольцо должно быть разомкнуто;
- сложность конфигурирования и настройки;
- сложность поиска неисправностей.

Кольцевая топология сети используется довольно редко. Основное применение она нашла в оптоволоконных сетях стандарта TokenRing.

Топология “звезда”

Звезда – это топология локальной сети, где каждая рабочая станция присоединена к центральному устройству (коммутатору или маршрутизатору) (рис.4). Центральное устройство управляет движением пакетов в сети. Каждый компьютер через сетевую карту подключается к коммутатору отдельным кабелем.

Рисунок 4. Топология звезда.

При необходимости можно объединить вместе несколько сетей с топологией “звезда” – в результате вы получите конфигурацию сети с ***древовидной*** топологией.

Древовидная топология распространена в крупных компаниях. Мы не будем ее подробно рассматривать в данной статье.

Топология “звезда” на сегодняшний день стала основной при построении локальных сетей. Это произошло благодаря ее многочисленным достоинствам:

- выход из строя одной рабочей станции или повреждение ее кабеля не отражается на работе всей сети в целом;
- отличная масштабируемость: для подключения новой рабочей станции достаточно проложить от коммутатора отдельный кабель;
- легкий поиск и устранение неисправностей и обрывов в сети;
- высокая производительность;
- простота настройки и администрирования;
- в сеть легко встраивается дополнительное оборудование.
-

Однако, как и любая топология, “звезда” не лишена недостатков:

- выход из строя центрального коммутатора обернется неработоспособностью всей сети;
- дополнительные затраты на сетевое оборудование – устройство, к которому будут подключены все компьютеры сети (коммутатор);
- число рабочих станций ограничено количеством портов в центральном коммутаторе.

Звезда – самая распространенная топология для проводных и беспроводных сетей. Примером звездообразной топологии является сеть с кабелем типа витая пара, и коммутатором в качестве центрального устройства. Именно такие сети встречаются в большинстве организаций.

3.2 Практическое занятие № 2 (2 часа)

Тема: «Коммутация»

3.2.1 Задачи для работы:

1.Ознакомиться со способами коммутации;

2. Изучить разделение каналов по времени и по частоте.

3.2.2 Краткое описание практического занятия:

Способы коммутации

Разделение каналов по времени

Разделение каналов по частоте

3.2.3 Результаты и выводы:

1. Способы коммутации.

Пакеты в сети могут передаваться двумя способами (рис.5):

- дейтаграммным;
- путем формирования «виртуального канала».

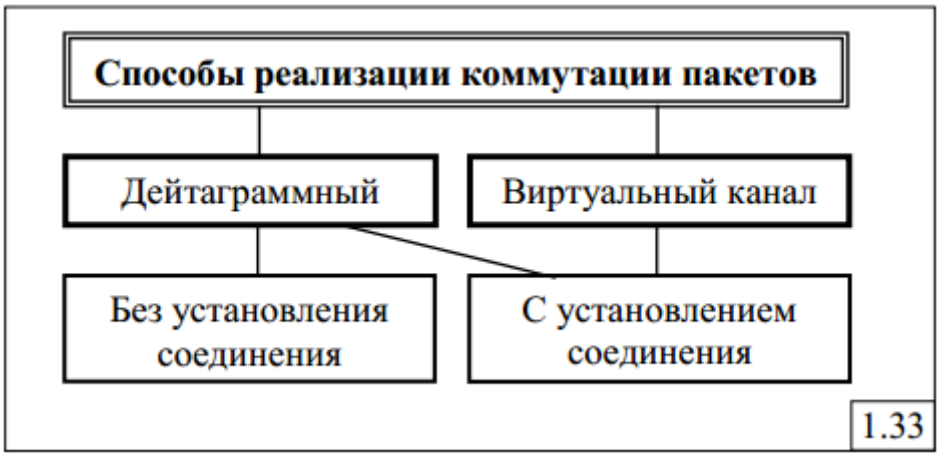


Рисунок 5. Способы реализации коммутации.

При **дейтаграммном** способе пакеты одного и того же сообщения могут передаваться между двумя взаимодействующими пользователями А и В по разным маршрутам, как это показано на рис.6, где пакет П1 передаётся по маршруту У1-У2-У6-У7, пакет П2 – по маршруту У1-У4-У7 и пакет П3 – по маршруту У1-У3-У5-У7.

В результате такого способа передачи все пакеты приходят в конечный узел сети в разное время и в произвольной последовательности. Пакеты одного и того же сообщения, рассматриваемые в каждом узле сети как самостоятельные независимые единицы данных и передаваемые разными маршрутами, называются дейтаграммами (datagram). В узлах сети для каждой дейтаграммы всякий раз определяется наилучший путь передачи в соответствии с выбранной метрикой маршрутизации, не зависимо от того, по какому пути переданы были предыдущие дейтаграммы с такими же адресами назначения (получателя) и источника (отправителя).

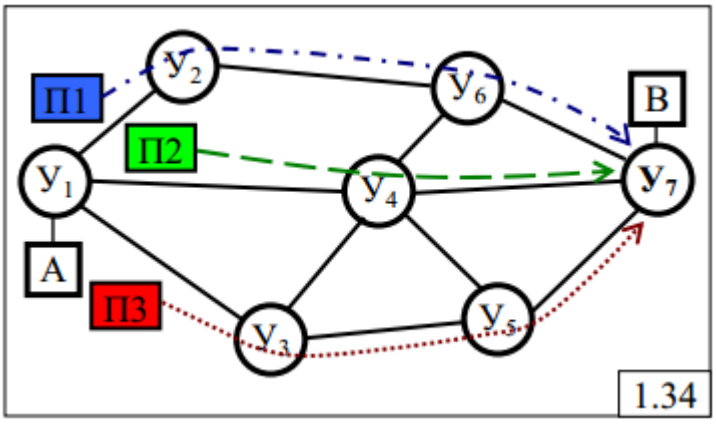


Рисунок 6. Дейтаграммный способ.

Дейтаграммный способ передачи пакетов может быть реализован:

- без установления соединения между абонентами сети;
- с установлением соединения между взаимодействующими абонентами сети.

В последнем случае между взаимодействующими абонентами предварительно устанавливается соединение путём обмена служебными пакетами: «запрос на соединение» и «подтверждение соединения», означающее готовность принять передаваемые данные. В процессе установления соединения могут «оговариваться» значения параметров передачи данных, которые должны выполняться в течение сеанса

связи. После установления соединения отправитель начинает передачу, причём пакеты одного и того же сообщения могут передаваться разными маршрутами, то есть дейтаграммным способом. По завершении сеанса передачи данных выполняется процедура разрыва соединения путём обмена служебными пакетами: «запрос на разрыв соединения» и «подтверждение разрыва соединения». Описанная процедура передачи пакетов с установлением соединения иллюстрируется на диаграмме (рис.7).

Достоинствами дейтаграммного способа передачи пакетов в компьютерных сетях являются:

- простота организации и реализации передачи данных – каждый пакет (дейтаграмма) сообщения передаётся независимо от других пакетов;
- в узлах сети для каждого пакета выбирается наилучший путь (маршрут);
- передача данных может выполняться как без установления соединения между взаимодействующими абонентами, так и при необходимости с установлением соединения.

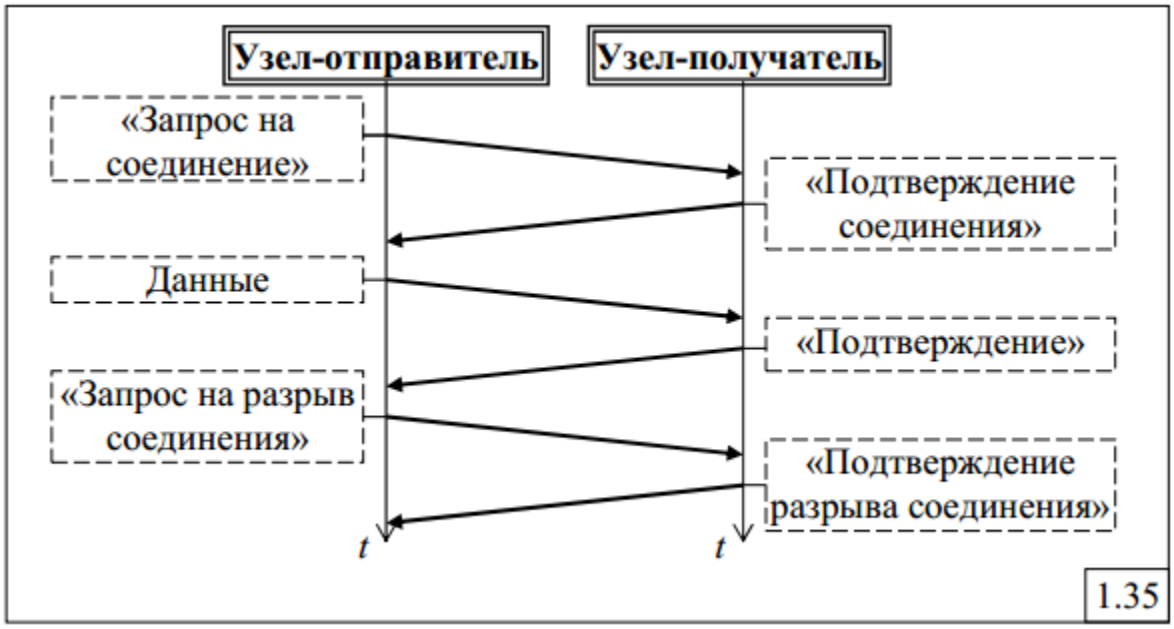


Рисунок 7. Процедура передачи пакетов

К недостаткам дейтаграммного способа передачи пакетов следует отнести:

- необходимость сборки сообщения в конечном узле: сообщение не может быть передано получателю, пока в конечном узле сети не соберутся все пакеты данного сообщения, поэтому в случае потери хотя бы одного пакета сообщение не сможет быть сформировано и передано получателю;
- при длительном ожидании пакетов одного и того же сообщения в конечном узле может скопиться достаточно большое количество пакетов сообщений, собранных не полностью, что требует значительных затрат на организацию в узле буферной памяти большой ёмкости;
- для предотвращения переполнения буферной памяти узла время нахождения (ожидания) пакетов одного и того же сообщения в конечном узле ограничивается, и по истечении этого времени все поступившие пакеты не полностью собранного сообщения уничтожаются, после чего выполняется запрос на повторную передачу данного сообщения; это приводит к увеличению нагрузки на сеть и, как следствие, к снижению её производительности, измеряемой количеством сообщений, передаваемых в сети за единицу времени.

Способ передачи пакетов «**виртуальный канал**» заключается в формировании единого «виртуального» канала на время взаимодействия абонентов для передачи всех пакетов сообщения. Этот способ реализуется с использованием предварительного установления соединения между взаимодействующими абонентами, в процессе которого формируется наиболее рациональный единый для всех пакетов маршрут, по которому, в отличие от дейтаграммного способа, все пакеты сообщения передаются в естественной последовательности, как это показано на рис.8.

Пакеты П1, П2 и П3 сообщения передаются в естественной последовательности от пользователя А к пользователю В по предварительно созданному виртуальному каналу через узлы У1-У4-У7.

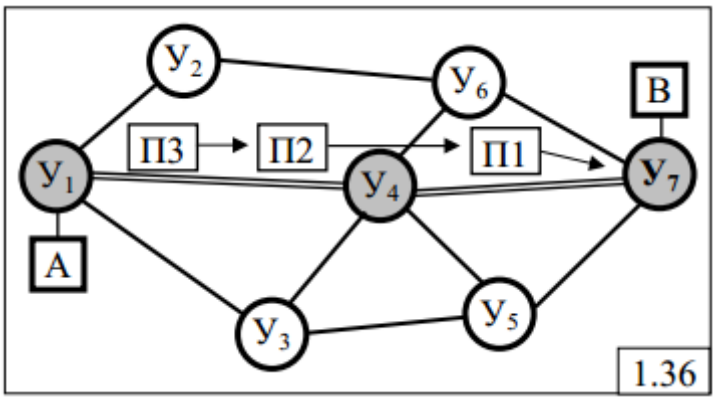


Рисунок 8. Передача пакетов сообщения.

Виртуальный канал, как и реальный физический канал в случае коммутации каналов, существует только в течение сеанса связи, при этом ресурсы реальных каналов связи (пропускная способность) и узлов сети (буферная память), находящихся на маршруте, резервируются на всё время сеанса.

Не следует путать и смешивать коммутацию каналов и способ передачи пакетов «виртуальный канал». Основное их отличие состоит в том, что «виртуальной канал» реализуется с промежуточным хранением пакетов в узлах сети, в то время как коммутация каналов реализуется без промежуточного хранения передаваемых пакетов за счёт создания реального (а не виртуального) физического канала между абонентами сети.

К достоинствам способа передачи пакетов «виртуальный канал» по сравнению с дейтаграммной передачей пакетов можно отнести:

- меньшие задержки в узлах сети, обусловленные резервированием ресурсов, и прежде всего пропускной способности каналов связи, в процессе установления соединения;
- небольшое время ожидания в конечном узле для сборки всего сообщения, поскольку пакеты передаются последовательно друг за другом по одному и тому же маршруту (виртуальному каналу), и вероятность того, что какой-либо пакет «заблудится» в результате неудачно выбранного маршрута или его время доставки окажется слишком большим, как это может произойти при дейтаграммном способе, близка к нулю;
- более эффективное использование буферной памяти промежуточных узлов за счёт её предварительного резервирования, а также буферной памяти в конечном узле в связи с небольшим временем ожидания прихода всех пакетов сообщения.

К недостаткам способа передачи пакетов «виртуальный канал» можно отнести:

- наличие накладных расходов (издержек) на установление соединения;

• неэффективное использование ресурсов сети, поскольку они резервируются на всё время взаимодействия абонентов (сеанса) и не могут быть предоставлены другому соединению, даже если они в данный момент не используются.

2. Разделение каналов по времени.

Коммутаторы, а также соединяющие их каналы должны обеспечивать одновременную передачу данных нескольких абонентских каналов. Для этого они должны быть высокоскоростными и поддерживать какую-либо технику мультиплексирования абонентских каналов. В настоящее время для мультиплексирования абонентских каналов используются две техники: техника частотного мультиплексирования (Frequency Division Multiplexing, FDM); техника мультиплексирования с разделением времени (Time Division Multiplexing, TDM).

Коммутация каналов на основе частотного мультиплексирования

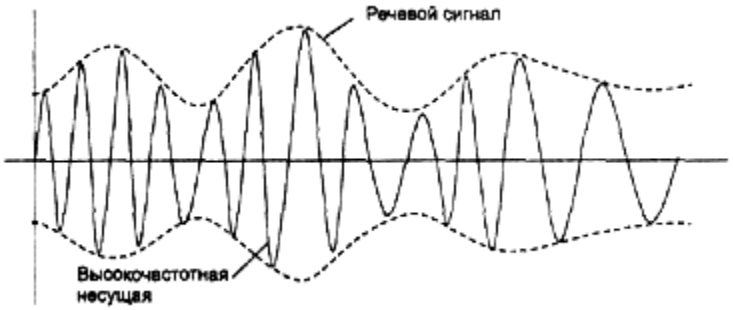


Рисунок 9. Модуляция речевых сигналов

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

Для разделения абонентских каналов характерна техника модуляции высокочастотного несущего синусоидального сигнала низкочастотным речевым сигналом (рис. 9). Если сигналы каждого абонентского канала перенести в свой собственный диапазон частот, то в одном широкополосном канале можно одновременно передавать сигналы нескольких абонентских каналов. На входы FDM- коммутатора поступают исходные сигналы от абонентов телефонной сети. Коммутатор выполняет перенос частоты каждого канала в свой диапазон частот. Обычно высокочастотный диапазон делится на полосы, которые отводятся для передачи данных абонентских каналов. Такой канал называют уплотненным. Коммутаторы FDM могут выполнять как динамическую, так и постоянную ком¬мутацию. При динамической коммутации один абонент инициирует соединение с другим абонентом, посылая в сеть номер вызываемого абонента. Коммутатор дина¬мически выделяет данному абоненту одну из свободных полос своего уплотненно¬го канала. При постоянной коммутации за абонентом полоса закрепляется на длительный срок путем настройки коммутатора по отдельному входу, недоступ¬ному пользователям. Коммутация каналов на основе разделения времени разрабатывалась в расчете на передачу непрерывных сигналов, представляющих голос. При переходе к цифро¬вой форме представления голоса была разработана новая техника мультиплекси¬рования, ориентирующаяся на дискретный характер передаваемых данных. Эта техника носит название мультиплексирования с разделением времени (Time Division Multiplexing, TDM). Реже используется и другое ее название — техника синхронного режима передачи (Synchronous Transfer Mode, STM). Аппаратура TDM-сетей — мультиплексоры, коммутаторы, демультиплексоры — работает в режиме разделения

времени, поочередно обслуживая в течение цикла своей работы все абонентские каналы. Цикл работы оборудования TDM равен 125 мкс, что соответствует периоду следования замеров голоса в цифровом абонентском канале. Это значит, что мультиплексор или коммутатор успевает вовремя обслужить любой абонентский канал и передать его очередной замер далее по сети. Каждому соединению выделяется один квант времени цикла работы аппаратуры, называемый также тайм-слотом. Длительность тайм-слота зависит от числа абонентских каналов, обслуживаемых мультиплексором TDM или коммутатором. Мультиплексор принимает информацию по N входным каналам от конечных абонентов, каждый из которых передает данные по абонентскому каналу со скоростью 64 Кбит/с — 1 байт каждые 125 мкс. В каждом цикле мультиплексор выполняет следующие действия: прием от каждого канала очередного байта данных; составление из принятых байтов уплотненного кадра, называемого также обоймой; передача уплотненного кадра на выходной канал с битовой скоростью, равной Nx64 Кбит/с. Порядок байт в обойме соответствует номеру входного канала, от которого этот байт получен. Количество обслуживаемых мультиплексором абонентских каналов зависит от его быстродействия. Демультиплексор выполняет обратную задачу — он разбирает байты уплотненного кадра и распределяет их по своим нескольким выходным каналам, при этом он считает, что порядковый номер байта в обойме соответствует номеру выходного канала.

Коммутатор принимает уплотненный кадр по скоростному каналу от мульти¬плексора и записывает каждый байт из него в отдельную ячейку своей буферной памяти, причем в том порядке, в котором эти байты были упакованы в уплотненный кадр. Для выполнения операции коммутации байты извлекаются из буферной памяти не в порядке поступления, а в таком порядке, который соответствует под¬держиваемым в сети соединениям абонентов.

Развитием идей статистического мультиплексирования стала технология асинхронного режима передачи — ATM, которая вобрала в себя лучшие черты техники коммутации каналов и пакетов.

3.Разделение каналов по частоте

Частотное разделение каналов (ЧРК) — разделение каналов по частоте, при котором каждому каналу выделяется определённый диапазон частот. В многоканальных системах связи (МКС) с ЧРК канальные сигналы отличаются друг от друга положением своих спектров на оси частот. Обычно системы с ЧРК используются для передачи аналоговых сигналов. На рис. 10 представлена структурная схема простейшей МКС с ЧРК.

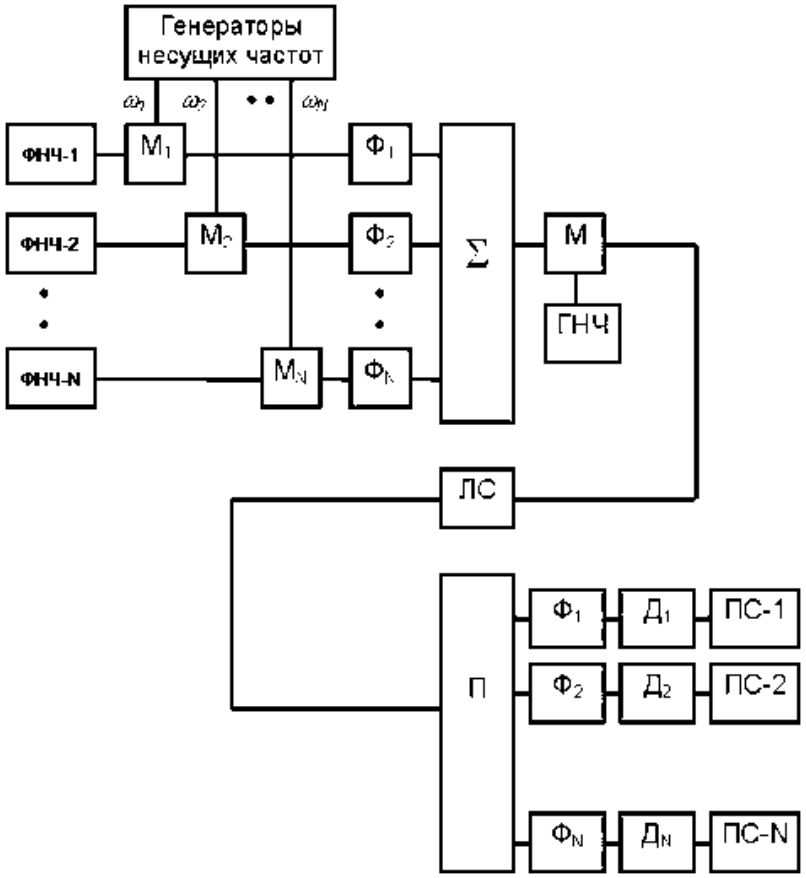


Рисунок 10. Структурная схема многоканальной системы с ЧРК

Наиболее часто в отдельных каналах при ЧРК применяется однополосная модуляция с соответственно подобранными частотами пилот-сигналов, которые выдаются генератором несущих частот (ГНЧ). Данный способ модуляции обеспечивает минимальную полосу частот группового сигнала. Подавление несущих достигается в индивидуальных модуляторах ($М_1$, $М_2$. ... $М_N$), которые, как правило, строятся по балансной схеме, а выделение одной боковой полосы (ОБП) осуществляется в полосовых фильтрах ($Ф_1$, $Ф_2$, ... $Ф_N$). Совокупность канальных сигналов на выходе суммирующего устройства Σ образует групповой сигнал. В групповом передатчике М групповой сигнал преобразуется в линейный сигнал, который и поступает в линию связи ЛС.

На приемной стороне линии связи линейный сигнал с помощью группового приемника П вновь преобразуется в групповой сигнал, из которого полосовыми фильтрами ($Ф_1$, $Ф_2$, ... $Ф_N$) выделяются канальные сигналы. После детектирования в канальных демодуляторах ($Д_1$, $Д_2$, ... $Д_N$) сигналы преобразуются в предназначенные получателям сообщения приемниками сообщений ($ПС_1$, $ПС_2$, ... $ПС_N$). Опорные колебания в канальных демодуляторах создаются с помощью генератора (ГНЧ). Сообщения, передаваемые по различным каналам, выделяются при помощи ФНЧ. .

Канальные передатчики вместе с суммирующим устройством образуют аппаратуру объединения. Групповой передатчик М, линия связи ЛС и групповой приемник П составляют групповой канал связи (тракт передачи), который вместе с аппаратурой объединения и индивидуальными приемниками составляет систему многоканальной связи. В составе технических устройств на передающей стороне многоканальной системы должна быть предусмотрена аппаратура объединения, а на приемной стороне - аппаратура разделения.

В общем случае групповой сигнал может формироваться не только простейшим суммированием канальных сигналов, но также и определенной логической обработкой, в

результате которой каждый элемент группового сигнала несет информацию о сообщениях источников. Это так называемые системы с комбинационным разделением.

Чтобы разделяющие устройства были в состоянии различать сигналы отдельных каналов, должны существовать определенные признаки, присущие только данному сигналу. Такими признаками в общем случае могут быть параметры переносчика, например амплитуда, частота или фаза в случае непрерывной модуляции гармонического переносчика. При дискретных видах модуляции различающим признаком может служить и форма сигналов. Соответственно различаются и способы разделения сигналов: частотный, временной, фазовый и др.

Поскольку всякая реальная линия связи обладает ограниченной полосой пропускания, то при многоканальной передаче каждому отдельному каналу отводится определенная часть общей полосы пропускания.

На приемной стороне одновременно действуют сигналы всех каналов, различающиеся положением их частотных спектров на шкале частот. Чтобы без взаимных помех разделить такие сигналы, приемные устройства должны содержать частотные фильтры. Каждый из фильтров должен пропустить без ослабления лишь те частоты, которые принадлежат сигналу данного канала; частоты сигналов всех других каналов фильтр должен подавить.

Для снижения переходных помех до допустимого уровня приходится вводить защитные частотные интервалы (Рис. 11)

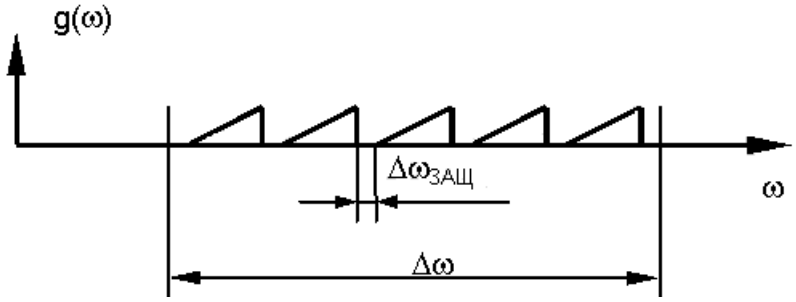


Рисунок 11. Спектр группового сигнала с защитными интервалами

При однократном преобразовании спектра сигнала (***fн = 104 кГц***) необходим специальный фильтр. При двукратном преобразовании спектра в первом модуляторе выбирается несущая до 30 кГц, например, ***f1 = 12 кГц***. При этом фильтр легко реализуется на ***LC***-элементах. На второй модулятор сигнал подается уже в полосе частот ***12,3…15,4 кГц***, и для переноса этого сигнала в заданную полосу частот необходимо использовать несущую ***f2 = 104 - f1 = 92 кГц***. Фильтр второго преобразователя частоты также легко реализуется на ***LC***-элементах.

Методы построения многоканальной аппаратуры с ЧРК отличаются способом формирования группового сигнала и особенностями передачи его в линейном тракте. По способу формирования группового сигнала (первый признак) различают:
1. метод с индивидуальным преобразованием сигналов;
2. метод с групповым преобразованием сигналов.
По способу усиления группового (линейного) сигнала (второй признак) выделяют:
1. метод с усилением каждого индивидуального сигнала;
2. метод с усилением линейного сигнала в целом.

При индивидуальном преобразовании сигналов формирование группового (линейного) спектра частот производится путем отдельного независимого преобразования каждого из ***N*** сигналов. Другими словами, при индивидуальном методе преобразователи, фильтры, усилители и другие элементы для каждого канала являются отдельными и повторяются в составе оконечной промежуточной аппаратуры столько раз, на сколько каналов рассчитана система передачи. Индивидуальные методы преобразования в оконечных и усиления в промежуточных станциях поясняются на рис. 12.

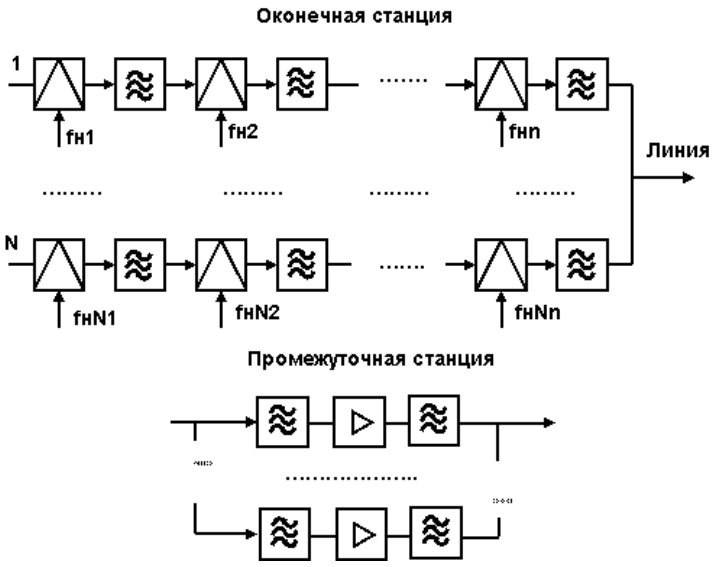


Рисунок 12. Индивидуальный метод преобразования

В основу метода группового преобразования сигналов положен принцип формирования линейного спектра в оконечном пункте МКС с помощью нескольких ступеней преобразования. На каждой ступени объединяются несколько канальных сигналов, т.е. линейный сигнал представляет собой сумму нескольких промежуточных групповых сигналов. При этом, в отличие от индивидуального метода, отдельной для каждого канала является только часть аппаратуры, а остальное оборудование – общее для всех каналов.

3.3 Практическое занятие № 3 (2 часа)
Тема: «Основные характеристики ЭВМ»
3.3.1 Задачи работы:
1.Преобразование текстовых данных при вводе в ЭВМ.
2.Преобразование растровой графики при вводе в ЭВМ.
3.Расчет размеров файлов данных при их различном представлении.
3.3.2 Краткое описание практического:
Преобразование текстовых данных при вводе в ЭВМ.
Преобразование растровой графики при вводе в ЭВМ.
Расчет размеров файлов данных при их различном представлении.
3.3.3Результаты и выводы:

1. ПРЕОБРАЗОВАНИЕ ТЕКСТОВЫХ ДАННЫХ ПРИ ВВОДЕ В ЭВМ

Представление данных заключается в их преобразовании в вид, удобный для последующей обработки либо пользователем, либо ЭВМ.

Форма представления данных определяется их конечным предназначением. В зависимости от этого данные имеют внутреннее и внешнее представление. Внутреннее представление данных (для ЭВМ) определяется физическими принципами, по которым происходит обмен сигналами между узлами ЭВМ, принципами организации памяти, логикой работы ЭВМ. Внутреннее представление данных в большинстве современных ЭВМ является дискретным, т.е. цифровым, причем любые данные для обработки ЭВМ

представляются последовательностями двух целых чисел – единицы и нуля. Такая форма представления данных получила название двоичной.

Во внешнем представлении (для пользователя) все данные хранятся в виде файлов. Файл – область памяти на внешнем носителе, которой присвоено имя.

Правило представления символьной информации (буквы алфавита и др. символы) заключается в том, что каждому символу ставится в соответствие уникальное число, т.е. каждый символ нумеруется.

Наиболее простой стандарт кодировки символов ASCII-код («Американский Стандартный Код для Обмена Информацией» – англ.AmericanStandartCodeforInformationInterchange) был введён в США еще в 1963 г. и после модификации в 1977 г. был принят в качестве всемирного стандарта. Каждому символу поставлено в соответствие двоичное число от 0 до 255 (8-битовый двоичный код). Символы от 0 до 127 – специальные символы (0-31), латинские буквы, цифры и знаки препинания составляют постоянную (базовую) часть таблицы. Расширенная таблица с 128 по 255 символ отводится под национальный стандарт. Пример расширенной таблицы представлен на рисунке 1, где код символа записан сокращенно в виде шестнадцатеричного числа, например, символуБсоответствует шестнадцатеричный кодC1 или в двоичной форме11000001.

Таким образом, каждый введенный в компьютер с клавиатуры или другим образом символ запоминается и хранится на носителе в виде восьмиразрядного двоичного слова. Текстовый файл в этом случае представляет собой последовательность байт данных, за каждым из которых стоит символ текста. При выводе текста на экран или на принтер соответствующие программно-аппаратные средства вывода выполняют обратную перекодировку из цифровой формы в символьную по тем же правилам.

Таблица 1. Представление чисел в различных системах счисления

Система счисления			
Десятичная	Двоичная	Восьмеричная	Шестнадцатеричная
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7

8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0				©	Ё	§	Є	·		°						
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1																
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2		!	"	#	$	%	&		(	)	*	+				/
	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	‘	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}]	~	□
	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	Ђ	Ѓ	‚	ѓ	„	…	†	‡		‰	Љ	‹	Њ	Ќ	Ћ	Џ
	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	ђ	‘	’	"	"		–	—		™	љ	›	њ	ќ	ћ	џ
	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A		Ў	ў	J	¤	Ґ	¦	§	Ё	©	Є	«		-	®	Ї
	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	°	±	І	і	ґ	µ		·	ё	№	є	»	j	S	s	ї
	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	▪ 223
E	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Рис. 1. Расширенная таблица ASCII кодов (кириллица windows 1251).

Более универсальным и мультиязычным является стандарт Unicode, который определяет кодировку каждого символа не одним байтом, а двумя. Соответственно, число одновременно кодируемых символов возрастает с 256 до 65536. Данный стандарт позволяет закодировать одновременно все известные символы, в том числе японские и китайские иероглифы.

2.ПРЕОБРАЗОВАНИЕ РАСТРОВОЙ ГРАФИКИ ПРИ ВВОДЕ В ЭВМ

Как и любые другие виды данных, графические данные в ЭВМ хранятся, обрабатываются и передаются в закодированном двоичном коде.

Существуют два принципиально разных подхода к представлению (оцифровке) графических данных: растровый и векторный.

Для оцифровки графических изображений при растровом представлении вся область данных разбивается на множество точечных элементов – **пикселей,** каждый из которых имеет свой цвет. Совокупность пикселей называется**растром**, а изображения, которые формируются на основе растра, называются**растровыми.**

Число пикселей по горизонтали и вертикали изображения определяет **разрешение** изображения**.**Каждый пиксель нумеруется, начиная с нуля, слева направо и сверху вниз. Пример представления треугольной области растровым способом показан на рис. 2.

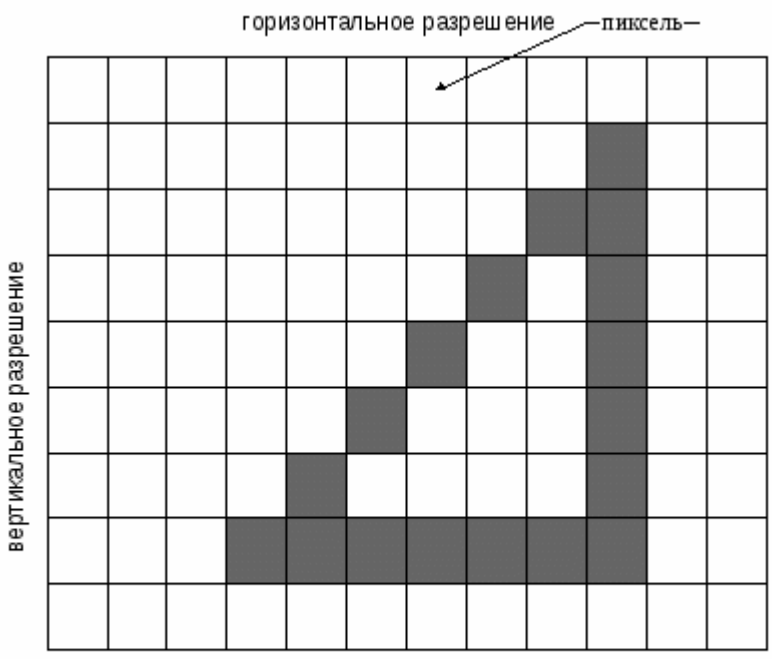


Рис. 2. Растровое изображение

При растровом способе представления графических данных под каждый пиксель отводится определенное число бит, называемое битовой глубиной и используемой для кодировки цвета пикселя. Каждому цвету соответствует определенный двоичный код (т.е. код из нулей и единиц).

Например, если битовая глубина равна 1, то под каждый пиксель отводится 1 бит. В этом случае 0 соответствует черному цвету, 1 – белому, а изображение может быть только черно-белым. Если битовая глубина равна 4, то каждый пиксель может быть закодирован цветовой гаммой из 16 цветов (2^4). При битовой глубине 8 каждый пиксель кодируется одним байтом, при этом количество цветов – 256. Вполне естественно, что с увеличением глубины цвета увеличивается объем памяти, необходимой для хранения графических данных.

Одним из распространенных способов кодирования цвета в растровой графике – модель RGB(R-Red,G-Green,B-Blue), в соответствии с которой произвольный оттенок цвета является суммой трех базовых цветов красного, зеленого и синего с разной интенсивностью. Интенсивность каждого базового цвета кодируется одним байтом, поэтому общее количество кодируемых оттенков цвета в моделиRGBсоставляет $2^{24} \approx$ 16.7 млн.

Таким образом, в растровой графике кодируется цвет каждого пикселя.

3. РАСЧЕТ РАЗМЕРОВ ФАЙЛОВ ДАННЫХ ПРИ ИХ РАЗЛИЧНОМ ПРЕДСТАВЛЕНИИ

Размер цифрового текстового файла определяется выражением:

$RF_{текст}$(в килобайтах) =$N_{сим}$*$К_{байт}$/1024,

где

$N_{сим}$- количество закодированных символов;

$К_{байт}$- количество байт для кодирования одного символа.

Размер растрового графического файла определяется выражением:

$RF_{графика}$(в мегабайтах) =$N_г$* $М_в К_{байт}$/1024,

где

$N_г$- количество пикселей по горизонтали;

$М_в$- количество пикселей по вертикали;

$К_{байт}$- количество байт для кодирования цвета одного пикселя.

3.4 Практическое занятие № 4 (2 часа)
Тема: «Минимальная конфигурация ЭВМ»
3.4.1 Задачи работы:
1.Ознакомится с минимальными конфигурациями ЭВМ
3.4.2 Краткое описание практического:
Конфигурации многоплатной микро ЭВМ
3.4.3Результаты и выводы:

Минимальная конфигурация многоплатной микро - ЭВМ базируется на телетайпный аппарат в качестве единственного стандартного внешнего устройства, обеспечивающего ввод-вывод перфоленты и документирование. Объем телетайпной версии диспетчерской системы - 1024 слова, что является минимальной конструктивной единицей при реализации ПЗУ на БИС.

Минимальная конфигурация, необходимая для работы ТМОС: процессор, ОП емкостью 8 Кслов, перфоленточное УВВ, консольный терминал. При расширении функций ТМОС используют ОП емкостью до 28 Кслов, АЦПУ, накопитель на магнитном диске кассетного типа.

Минимальная конфигурация АРМ включает персональный компьютер типа IBM PC и стандартный принтер.

Минимальная конфигурация оборудования: процессор СМ-3П или СМ-4П, оперативное запоминающее устройство емкостью не менее 16К слов, таймер, консольный терминал, дисковое ЗУ, пер-фоленточное устройство ввода-вывода, аппаратный начальный заг - (рузчик.

Минимальная конфигурация ВК - два модуля; в этом случае производительность ЭТК-КС составляет 0 17 сообщ. Для максимальной конфигурации ВК производительность равна 0 7 сообщ.

Минимальная конфигурация ЭВМ для функционирования СР / М-86 включает следующие устройства: процессор с ОЗУ емкостью 128 Кбайт, клавиатуру, дисплей, один НГМД.

Минимальная конфигурация ПЭВМ, необходимая для функционирования пакета: ОЗУ с емкостью не менее 320 Кбайт; один НГМД; графический дисплей с разрешающей способностью 640Х200 точек; печатающее устройство с возможностями вывода графиков.

Минимальная конфигурация ТС: процессор ОП (16 Кслов или 24 Кслов в зависимости от типа процессора); внешняя память на НМД ИЗОТ 1370, таймер, консоль. Диспетчер памяти расширяет программную адресацию ОП с 28 до 124 Кслов и обеспечивает защиту памяти. Версии ОС РВ, использующие ДП (ОС РВ с ДП), требуют минимально 24 Кслов ОП и обеспечивают работу многих пользователей в конфигурациях комплекса емкостью ОП до 121 Кслов. Версии ОС РВ, не использующие возможности ДП (ОС РВ без ДП), допускают конфигурации оборудования комплекса с ОП от 16 до 28 Кслов.

Минимальная конфигурация ТС: процессор типа СМ-4П; ОП емкостью 16 Колов (при этом обеспечивается одновременная работа двух пользователей); НМД с фиксированными головками или кассетного тина; НМЛ; таймер; перфоленточное УВВ; консоль; АЦПУ.

Минимальная конфигурация ЕС ЭВМ - обязательный набор технических средств ЕС ЭВМ, позволяющий реализовать вычислительный процесс средствами математического обеспечения.

Понятие *минимальная конфигурация ПЭВМ* (ПК) обычно связывается с конкретным типом центрального процессора, стандартными или минимальными для него размерами внутренней и внешней памяти, клавиатурой и монитором. [11]

В *минимальной конфигурации* программно каждый проект должен состоять из главного файла проекта и одного или нескольких модулей. [12]

Режим *минимальной конфигурации* (MN / MX 1) позволяет без дополнительных внешних средств создавать МС простого вида. В этом режиме все необходимые сигналы управления МП вырабатывает сам. [13]

В *минимальной конфигурации* этот абонентский пункт представляет собой компактное устройство, которое состоит из микро - ЭВМ К-1520; печатающего устройства последовательного действия СМ-6317, бесконтактной клавиатуры. [14]

В состав *минимальной конфигурации ПЭВМ* входят узлы, объединенные общей системной магистралью. Магистраль содержит три обязательные группы шин: данных, адреса и управления.

3.5 Практическое занятие № 5 (2 часа)
Тема: «Состав команд и архитектура 8086 микропроцессора»
3.5.1 Задачи работы:
1. Ознакомится с архитектурой микропроцессора 8086
2. Ознакомится с составом команд
3.5.2 Краткое описание практического:

Состав команд микропроцессора 8086
Архитектура микропроцессора 8086

3.5.3Результаты и выводы:

Обычно, когда новая архитектура создается одним архитектором или группой архитекторов, ее отдельные части очень хорошо подогнаны друг к другу и вся архитектура может быть описана достаточно связано. Этого нельзя сказать об архитектуре 80x86, поскольку это продукт нескольких независимых групп разработчиков, которые развивали эту архитектуру более 15 лет, добавляя новые возможности к первоначальному набору команд.

В 1978 году была анонсирована архитектура Intel 8086 как совместимое вверх расширение в то время успешного 8-бит микропроцессора 8080. 8086 представляет собой 16-битовую архитектуру со всеми внутренними регистрами, имеющими 16-битовую разрядность. Микропроцессор 8080 был просто построен на базе накапливающего сумматора (аккумулятора), но архитектура 8086 была расширена дополнительными регистрами. Поскольку почти каждый регистр в этой архитектуре имеет определенное назначение, 8086 по классификации частично можно отнести к машинам с накапливающим сумматором, а частично - к машинам с регистрами общего назначения, и его можно назвать расширенной машиной с накапливающим сумматором. Микропроцессор 8086 (точнее его версия 8088 с 8-битовой внешней шиной) стал основой завоевавшей в последствии весь мир серии компьютеров IBM PC, работающих под управлением операционной системы MS-DOS.

В 1980 году был анонсирован сопроцессор плавающей точки 8087. Эта архитектура расширила 8086 почти на 60 команд плавающей точки. Ее архитекторы отказались от расширенных накапливающих сумматоров для того, чтобы создать некий гибрид стеков и регистров, по сути расширенную стековую архитектуру. Полный набор стековых команд дополнен ограниченным набором команд типа регистр-память.

Анонсированный в 1982 году микропроцессор 80286, еще дальше расширил архитектуру 8086. Была создана сложная модель распределения и защиты памяти, расширено адресное пространство до 24 разрядов, а также добавлено небольшое число дополнительных команд. Поскольку очень важно было обеспечить выполнение без изменений программ, разработанных для 8086, в 80286 был предусмотрен режим реальных адресов, позволяющий машине выглядеть почти как 8086. В 1984 году компания IBM объявила об использовании этого процессора в своей новой серии персональных компьютеров IBM PC/AT.

В 1987 году появился микропроцессор 80386, который расширил архитектуру 80286 до 32 бит. В дополнение к 32-битовой архитектуре с 32-битовыми регистрами и 32-битовым адресным пространством, в микропроцессоре 80386 появились новые режимы адресации и дополнительные операции. Все эти расширения превратили 80386 в машину, по идеологии близкую к машинам с регистрами общего назначения. В дополнение к механизмам сегментации памяти, в микропроцессор 80386 была добавлена также поддержка страничной организации памяти. Также как и 80286, микропроцессор 80386 имеет режим выполнения программ, написанных для 8086. Хотя в то время базовой операционной системой для этих микропроцессоров оставалась MS-DOS, 32-разрядная архитектура и страничная организация памяти послужили основой для переноса на эту платформу операционной системы UNIX. Следует отметить, что для процессора 80286 была создана операционная система XENIX (сильно урезанный вариант системы UNIX).

Эта история иллюстрирует эффект, вызванный необходимостью обеспечения совместимости с 80x86, поскольку существовавшая база программного обеспечения на каждом шаге была слишком важной. К счастью, последующие процессоры (80486 в 1989 и Pentium в 1993 году) были нацелены на увеличение производительности и добавили к

видимому пользователем набору команд только три новые команды, облегчающие организацию многопроцессорной работы.

Что бы ни говорилось о неудобствах архитектуры 80x86, следует иметь в виду, что она преобладает в мире персональных компьютеров. Почти 80% установленных малых систем базируются именно на этой архитектуре. Споры относительно преимуществ CISC и RISC архитектур постепенно стихают, поскольку современные микропроцессоры стараются вобрать в себя наилучшие свойства обоих подходов.

Современное семейство процессоров i486 (i486SX, i486DX, i486DX2 и i486DX4), в котором сохранились система команд и методы адресации процессора i386, уже имеет некоторые свойства RISC-микропроцессоров. Например, наиболее употребительные команды выполняются за один такт. Компания Intel для оценки производительности своих процессоров ввела в употребление специальную характеристику, которая называется рейтингом iCOMP. Компания надеется, что эта характеристика станет стандартной тестовой оценкой и будет применяться другими производителями микропроцессоров, однако последние с понятной осторожностью относятся к системе измерений производительности, введенной компанией Intel. Ниже в таблице приведены сравнительные характеристики некоторых процессоров компании Intel на базе рейтинга iCOMP.

Процессор	Тактовая частота (МГц)	Рейтинг iCOMP
386SX386SL386DX386DXi486SXi486SXi486SXi486DXi486DX2i486DXi486DX2i486DX4i486DX4PentiumPentiumPentiumPentiumPentiumPentium	25252533202533335050667510060690100120133	3941496878100136166231249297319435510567735815100012 00

Процессоры i486SX и i486DX - это 32-битовые процессоры с внутренней кэш-па-мятью емкостью 8 Кбайт и 32-битовой шиной данных. Основное отличие между ними заключается в том, что в процессоре i486SX отсутствует интегрированный сопроцессор плавающей точки. Поэтому он имеет меньшую цену и применяется в системах, для которых не очень важна производительность при обработке вещественных чисел. Для этих систем обычно возможно расширение с помощью внешнего сопроцессора i487SX.

Процессоры Intel OverDrive и i486DX2 практически идентичны. Однако кристалл OverDrive имеет корпус, который может устанавливаться в гнездо расширения сопроцессора i487SX, применяемое в ПК на базе i486SX. В процессорах OverDrive и i486DX2 применяется технология удвоения внутренней тактовой частоты, что позволяет увеличить производительность процессора почти на 70%. Процессор i486DX4/100 использует технологию утроения тактовой частоты. Он работает с внутренней тактовой частотой 99 МГц, в то время как внешняя тактовая частота (частота, на которой работает внешняя шина) составляет 33 МГц. Этот процессор практически обеспечивает равные возможности с машинами класса 60 МГц Pentium, являясь их полноценной и доступной по цене альтернативой.

Появившийся в 1993 году процессор Pentium ознаменовал собой новый этап в развитии архитектуры x86, связанный с адаптацией многих свойств процессоров с архитектурой RISC. Он изготовлен по 0.8 микронной БиКМОП технологии и содержит 3.1 миллиона транзисторов. Первоначальная реализация была рассчитана на работу с тактовой частотой 60 и 66 МГц. В настоящее время имеются также процессоры Pentium, работающие с тактовой частотой 75, 90, 100 и 120 МГц. Процессор Pentium по сравнению со своими предшественниками обладает целым рядом улучшенных характеристик. Главными его особенностями являются:

- двухпотоковая суперскалярная организация, допускающая параллельное выполнение пары простых команд;

- наличие двух независимых двухканальных множественно-ассоциативных кэшей для команд и для данных, обеспечивающих выборку данных для двух операций в каждом такте;
- динамическое прогнозирование переходов;
- конвейерная организация устройства плавающей точки с 8 ступенями;
- двоичная совместимость с существующими процессорами семейства 80x86.

Блок-схема процессора Pentium представлена на рис. 8.1. Прежде всего новая микроархитектура этого процессора базируется на идее суперскалярной обработки (правда с некоторыми ограничениями). Основные команды распределяются по двум независимым исполнительным устройствам (конвейерам U и V). Конвейер U может выполнять любые команды семейства x86, включая целочисленные команды и команды с плавающей точкой. Конвейер V предназначен для выполнения простых целочисленных команд и некоторых команд с плавающей точкой. Команды могут направляться в каждое из этих устройств одновременно, причем при выдаче устройством управления в одном такте пары команд более сложная команда поступает в конвейер U, а менее сложная - в конвейер V. Такая попарная выдача команд возможна правда только для ограниченного подмножества целочисленных команд. Команды арифметики с плавающей точкой не могут запускаться в паре с целочисленными командами. Одновременная выдача двух команд возможна только при отсутствии зависимостей по регистрам. При остановке команды по любой причине в одном конвейере, как правило останавливается и второй конвейер.

Остальные устройства процессора предназначены для снабжения конвейеров необходимыми командами и данными. В отличие от процессоров i486 в процессоре Pentium используется раздельная кэш-память команд и данных емкостью по 8 Кбайт, что обеспечивает независимость обращений. За один такт из каждой кэш-памяти могут считываться два слова. При этом кэш-память данных построена на принципах двухкратного расслоения, что обеспечивает одновременное считывание двух слов, принадлежащих одной строке кэш-памяти. Кэш-память команд хранит сразу три копии тегов, что позволяет в одном такте считывать два командных слова, принадлежащих либо одной строке, либо смежным строкам для обеспечения попарной выдачи команд, при этом третья копия тегов используется для организации протокола наблюдения за когерентностью состояния кэш-памяти. Для повышения эффективности перезагрузки кэш-памяти в процессоре применяется 64-битовая внешняя шина данных.

В процессоре предусмотрен механизм динамического прогнозирования направления переходов. С этой целью на кристалле размещена небольшая кэш-память, которая называется буфером целевых адресов переходов (BTB), и две независимые пары буферов предварительной выборки команд (по два 32-битовых буфера на каждый конвейер). Буфер целевых адресов переходов хранит адреса команд, которые находятся в буферах предварительной выборки. Работа буферов предварительной выборки организована таким образом, что в каждый момент времени осуществляется выборка команд только в один из буферов соответствующей пары. При обнаружении в потоке команд операции перехода вычисленный адрес перехода сравнивается с адресами, хранящимися в буфере BTB. В случае совпадения предсказывается, что переход будет выполнен, и разрешается работа другого буфера предварительной выборки, который начинает выдавать команды для выполнения в соответствующий конвейер. При несовпадении считается, что переход выполняться не будет и буфер предварительной выборки не переключается, продолжая обычный порядок выдачи команд. Это позволяет избежать простоев конвейеров при правильном прогнозе направления перехода. Окончательное решение о направлении перехода естественно принимается на основании анализа кода условия. При неправильно сделанном прогнозе содержимое конвейеров аннулируется и выдача команд начинается с необходимого адреса. Неправильный прогноз приводит к приостановке работы конвейеров на 3-4 такта.

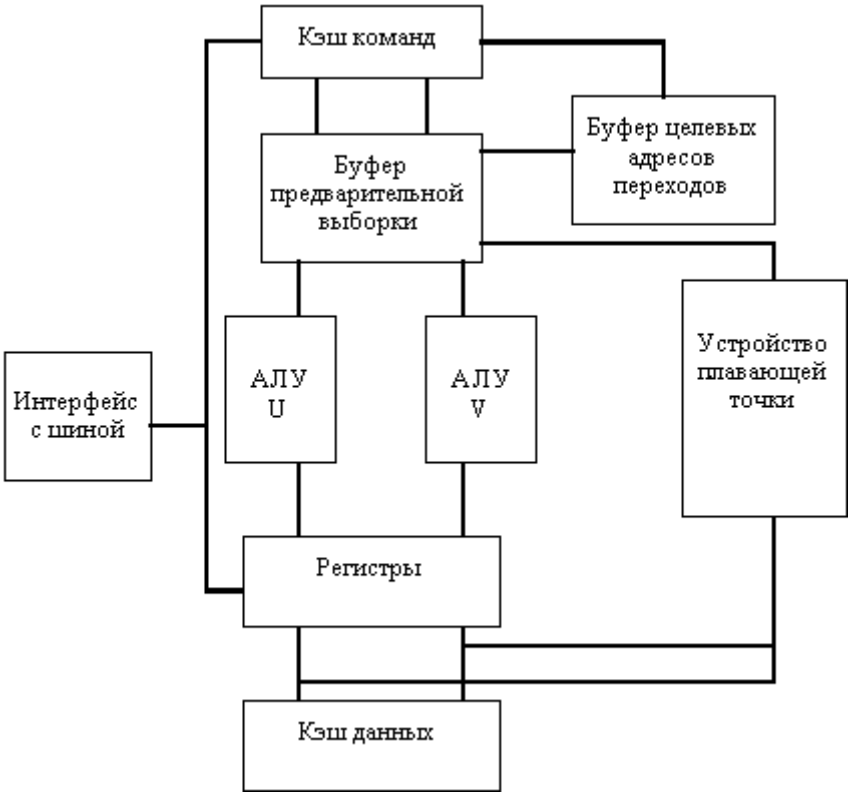


Рис. 8.1. Упрощенная блок схема процессора Pentium

Следует отметить, что возросшая производительность процессора Pentium требует и соответствующей организации системы на его основе. Компания Intel разработала и поставляет все необходимые для этого наборы микросхем. Прежде всего для согласования скорости с динамической основной памятью необходима кэш-память второго уровня. Контроллер кэш-памяти 82496 и микросхемы статической памяти 82491 обеспечивают построение такой кэш-памяти объемом 256 Кбайт и работу процессора без тактов ожидания. Для эффективной организации систем Intel разработала стандарт на высокопроизводительную локальную шину PCI. Выпускаются наборы микросхем для построения мощных компьютеров на ее основе.

В настоящее время компания Intel ведет разработку нового процессора, продолжающего архитектурную линию x86. Этот процессор получил название P6. По оценкам экспертов число транзисторов в новом кристалле составит 4-5 миллионов, что может обеспечить повышение производительности до уровня 200 MIPS (66 МГц Pentium имеет производительность 112 MIPS). Для достижения такой производительности необходимо использование технических решений, широко применяющихся при построении RISC-процессоров:

- выполнение команд не в предписанной программой последовательности, что устраняет во многих случаях приостановку конвейеров из-за ожидания операндов операций;
- использование методики переименования регистров, позволяющей увеличивать эффективный размер регистрового файла (малое количество регистров - одно из самых узких мест архитектуры x86);
- расширение суперскалярных возможностей по отношению к процессору Pentium, в котором обеспечивается одновременная выдача только двух команд с достаточно жесткими ограничениями на их комбинации.

Кроме того, в борьбу за новое поколение процессоров x86 включились компании, ранее занимавшиеся изготовлением Intel-совместимых процессоров. Это компании Advanced Micro Devices (AMD), Cyrix Corp и NexGen. С точки зрения микроархитектуры наиболее близок к Pentium процессор M1 компании Cyrix, который должен появиться на рынке в ближайшее время. Также как и Pentium он имеет два конвейера и может выполнять до

двух команд в одном такте. Однако в процессоре M1 число случаев, когда операции могут выполняться попарно, значительно увеличено. Кроме того в нем применяется методика обходов и ускорения пересылки данных, позволяющая устранить приостановку конвейеров во многих ситуациях, с которыми не справляется Pentium. Процессор содержит 32 физических регистра (вместо 8 логических, предусмотренных архитектурой x86) и применяет методику переименования регистров для устранения зависимостей по данным. Как и Pentium, процессор M1 для прогнозирования направления перехода использует буфер целевых адресов перехода емкостью 256 элементов, но кроме того поддерживает специальный стек возвратов, отслеживающий вызовы процедур и последующие возвраты.

Процессоры K5 компании AMD и Nx586 компании NexGen используют в корне другой подход. Основа их процессоров - очень быстрое RISC-ядро, выполняющее высокорегулярные операции в суперскалярном режиме. Внутренние форматы команд (ROP у компании AMD и RISC86 у компании NexGen) соответствуют традиционным системам команд RISC-процессоров. Все команды имеют одинаковую длину и кодируются в регулярном формате. Обращения к памяти выполняются специальными командами загрузки и записи. Как известно, архитектура x86 имеет очень сложную для декодирования систему команд. В процессорах K5 и Nx586 осуществляется аппаратная трансляция команд x86 в команды внутреннего формата, что дает лучшие условия для распараллеливания вычислений. В процессоре K5 имеются 40, а в процессоре Nx586 22 физических регистра, которые реализуют методику переименования. В процессоре K5 информация, необходимая для прогнозирования направления перехода, записывается прямо в кэш команд и хранится вместе с каждой строкой кэш-памяти. В процессоре Nx586 для этих целей используется кэш-память адресов переходов на 96 элементов.

Таким образом, компания Intel больше не обладает монополией на методы конструирования высокопроизводительных процессоров x86, и можно ожидать появления новых процессоров, не только не уступающих, но и возможно превосходящих по производительности процессоры компании, стоявшей у истоков этой архитектуры. Следует отметить, что сама компания Intel заключила стратегическое соглашение с компанией Hewlett-Packard на разработку следующего поколения микропроцессоров, в которых архитектура x86 будет сочетаться с архитектурой очень длинного командного слова (VLIW -архитектурой). Появление этих микропроцессоров не ожидается до конца 1998 года.

3.6 Практическое занятие № 6 (2 часа)
Тема: «Сетевое оборудование»
3.6.1 Задачи работы:
1. Ознакомится с сетевыми адаптерами
2. Ознакомиться с концентраторами
3. Ознакомиться с коммутаторами
4. Ознакомиться с маршрутизаторами

3.6.2 Краткое описание практического:
Сетевые адаптеры
Концентраторы
Коммутаторы
Маршрутизаторы

3.6.3Результаты и выводы:

1. Сетевые адаптеры.

Сетевой адаптер (Network Interface Card, NIC) - это периферийное устройство компьютера, непосредственно взаимодействующее со средой передачи данных, которая прямо или через другое коммуникационное оборудование связывает его с другими компьютерами. Это устройство решает задачи надежного обмена двоичными данными, представленными соответствующими электромагнитными сигналами, по внешним линиям связи. Как и любой контроллер компьютера, сетевой адаптер работает под управлением драйвера операционной системы и распределение функций между сетевым адаптером и драйвером может изменяться от реализации к реализации. Сетевые адаптеры и кабели являются аппаратной основой организации компьютерных сетей, их нормальная работа жизненно важна для сети. С кабелями и адаптерами связано обычно 80% неполадок в сети.

Сетевой адаптер вместе со своим драйвером реализует второй, канальный уровень модели открытых систем в конечном узле сети – компьютере.

Сетевой адаптер совместно с драйвером выполняет две операции: передачу и прием кадра.

В каждом компьютере должен быть установлен сетевой адаптер, обеспечивающий подключение к выбранному типу кабеля. Платы сетевого адаптера выступают в качестве физического интерфейса, или соединения между компьютером и сетевым кабелем. Платы вставляются в слоты расширения всех сетевых компьютеров и серверов.

В большинстве современных стандартов для локальных сетей предполагается, что между сетевыми адаптерами взаимодействующих компьютеров устанавливается специальное коммуникационное устройство (концентратор, мост, коммутатор или маршрутизатор), которое берет на себя некоторые функции по управлению потоком данных.

Сетевой адаптер обычно выполняет следующие функции:

• Формирование передаваемой информации в виде кадра определенного формата. Кадр включает несколько служебных полей, среди которых имеется адрес компьютера назначения и контрольная сумма кадра, по которой сетевой адаптер станции назначения делает вывод о корректности доставленной по сети информации.

• Получение доступа к среде передачи данных. В локальных сетях в основном применяются разделяемые между группой компьютеров каналы связи (общая шина, кольцо), доступ к которым предоставляется по специальному алгоритму (наиболее часто применяются метод случайного доступа или метод с передачей маркера доступа по кольцу).

• Кодирование последовательности бит кадра последовательностью электрических сигналов при передаче данных и декодирование при их приеме.

• Преобразование информации из параллельной формы в последовательную и обратно. Эта операция связана с тем, что для упрощения проблемы синхронизации сигналов и удешевления линий связи в вычислительных сетях информация передается в последовательной форме, бит за битом, а не побайтно, как внутри компьютера.

• Синхронизация битов, байтов и кадров. Для устойчивого приема передаваемой информации необходимо поддержание постоянного синхронизма приемника и передатчика информации. Сетевой адаптер использует для решения этой задачи специальные методы кодирования, не использующие дополнительной шины с тактовыми синхросигналами. Эти методы обеспечивают периодическое изменение состояния передаваемого сигнала, которое используется тактовым генератором приемника для подстройки синхронизма. Кроме синхронизации на уровне битов, сетевой адаптер решает задачу синхронизации и на уровне байтов, и на уровне кадров.

Функцией сетевого адаптера является передача и прием сетевых сигналов из кабеля. Адаптер воспринимает команды и данные от сетевой операционной системы (ОС), преобразует эту информацию в один из стандартных форматов и передает ее в сеть через подключенный к адаптеру кабель.

Сетевые адаптеры различаются также по типу принятой в сети сетевой технологии - Ethernet, Token Ring, FDDI и т.п. Как правило, конкретная модель сетевого адаптера работает по определенной сетевой технологии (например, Ethernet). В связи с тем, что для каждой технологии сейчас имеется возможность использования различных сред передачи данных (тот же Ethernet поддерживает коаксиальный кабель, неэкранированную витую пару и оптоволоконный кабель), сетевой адаптер может поддерживать как одну, так и одновременно несколько сред. В случае, когда сетевой адаптер поддерживает только одну среду передачи данных, а необходимо использовать другую, применяются трансиверы и конверторы.

2. Концентраторы.

Концентратор (hub) – это сетевое устройство, предназначенное для объединения устройств сети в сегменты. Основной принцип его работы заключается в трансляции пакетов, поступающих на один из его портов на все другие порты. Таким образом, пакет, поступивший в сеть, будет отправлен всем остальным устройствам сети, т.е. будет осуществляться широковещательная передача. Концентратор работает на физическом уровне модели взаимодействия открытых систем (OSI). Концентратор используется в различных технологиях: ATM, xDSL, Token Ring, но наибольшее распространение он нашел в технологии Ethernet.

Концентратор можно рассматривать как репитер с несколькими выходами. В отличие от switch он не анализирует содержимое пакетов или их заголовки, а просто копирует их. Hub не позволяет увеличить число устройств в одном сегменте или разгрузить его, уменьшив число коллизий. Основная его задача – это подключение новых устройств к сети и организация ее топологии. Кроме того, hub может быть использован для организации резервных каналов.

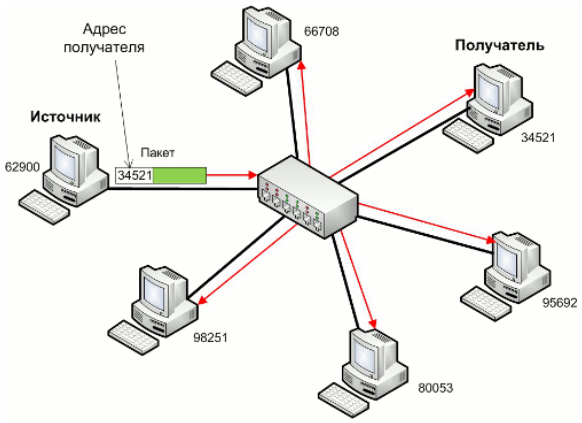


Рисунок 17. Пример работы сети с концентратором

Главным достоинством концентратора является простота реализации и, соответственно, невысокая стоимость. Однако из-за того, что он просто копирует пакеты во все свои порты, то в сети увеличивается вероятность возникновения коллизий. Это может привести к снижению скорости передачи и времени доставки пакетов. Именно поэтому вместо концентраторов обычно стараются применять коммутаторы, которые передают пакеты только к тому порту, к которому подключен компьютер получатель.

В зависимости от выполняемых задач можно встретить различные по емкости концентраторы от 4 до 64 портов. Однако это не предел. Они могут объединяться в более емкие устройства. Максимально возможное число работающих в спаренном режиме устройств ограничивается лишь характеристиками используемой технологии (для Ethernet – 1024 портов в одном сегменте). Концентраторы отличаются также по типу используемых проводников (витая пара, коаксиальный кабель) и используемой среде передачи (электрический или оптический кабель).

3. Коммутаторы.

Сетевой коммутатор (network switch) – это устройство, используемое в сетях передачи пакетов, предназначенное для объединения нескольких сегментов. В отличие от маршрутизатора (router) коммутатор работает на канальном уровне модели OSI, что и определяет главные различия между ними. Коммутатор не занимается расчетом маршрута для дальнейшей передачи пакетов по сети, анализируя различные факторы, как это делает маршрутизатор. Switch только передает данные от одного порта к другому на основе содержащейся в пакете информации. Обычно признаком выбора выходного порта служит MAC-адрес устройства, к которому передаются данные. В свою очередь коммутатор в отличие от концентратора или репитера не просто транслирует порты ко всем выходам, которые у него есть, а к одному, заранее выбранному.

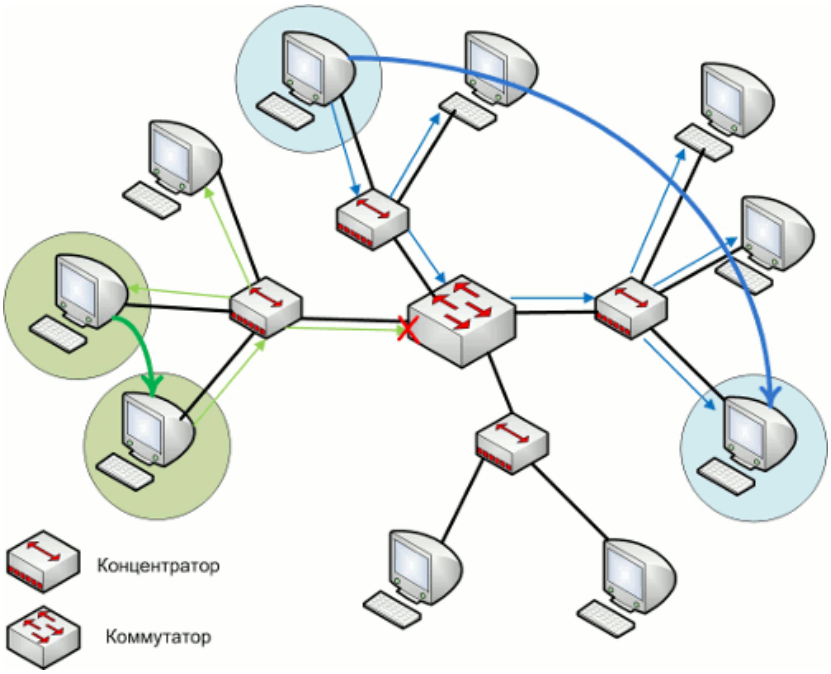


Рисунок 18. Пример сети с коммутатором

Сетевые коммутаторы применяются в нескольких технологиях, но наибольшее распространение нашли в Ethernet. Главной их задачей в сети Ethernet является разделение сети на сегменты. Это особенно актуально в сетях с большим числом рабочих станций, т.к. чем больше оконечных устройств работают одновременно с единой средой передачи данных тем выше вероятность возникновения коллизии (одновременной передачи данных несколькими устройствами) и, следовательно, ниже эффективность работы сети. Коммутатор позволяет разбить единую сеть на несколько сегментов и увеличить число одновременно работающих устройств.

Существую управляемые и неуправляемые коммутаторы. Неуправляемые коммутаторы самонастраиваются после включения в сеть. Они анализируют MAC-адреса всех устройств, подключенных к ним и будут осуществлять коммутацию между портами на основе анализа заголовка пакета, в котором содержится MAC-адресом устройства-получателя. Управляемые коммутаторы предоставляют интерфейс для администратора, который может выполнить его настройку для работы в конкретной сети. Например, есть возможность выбора режима защиты от отказа (в случае работы в паре с резервным коммутатором), объединения нескольких портов в единое направление, настройки приоритетов и резервирования портов и мн. др. Обычно управляемые коммутаторы дороже и используются в емких сетях, с дополнительными требованиями по надежности.

Switch может быть выполнен и в виде небольшой платы на 4 порта и многополочного штатива с возможность интеграции дополнительных устройств и расширения емкости. Также в зависимости от назначения сетевой коммутатор может снабжаться автономным питанием, портами управления и резервирования, охлаждением.

4. Маршрутизатор.

Маршрутизатор – это устройство пакетной сети передачи данных, предназначенное для объединения сегментов сети и ее элементов и служит для передачи пакетов между ними на основе каких-либо правил. Маршрутизаторы работают на сетевом (третьем) уровне модели OSI в качестве узловых устройств для различных технологий: IP, ATM, Frame Relay и мн. др.

Одной из самых важных задач маршрутизаторов является выбор оптимального маршрута передачи пакетов между подключенными сетями. Причем сделать это необходимо максимально оперативно с минимальной временной задержкой. Одновременно с этим должна отслеживаться текущая обстановка в сети для исключения из возможных путей доставки перегруженные и поврежденные участки. Практически все маршрутизаторы используют в своей работе, так называемые, таблицы маршрутизации. Это своеобразные базы данных, которые содержат информацию обо всех возможных маршрутах передачи пакетов с некоторой дополнительной информацией, которая берется в расчет при выборе оптимального варианта доставки. Это может быть состояние канала, время доставки информации, загруженность, полоса пропускания и др.

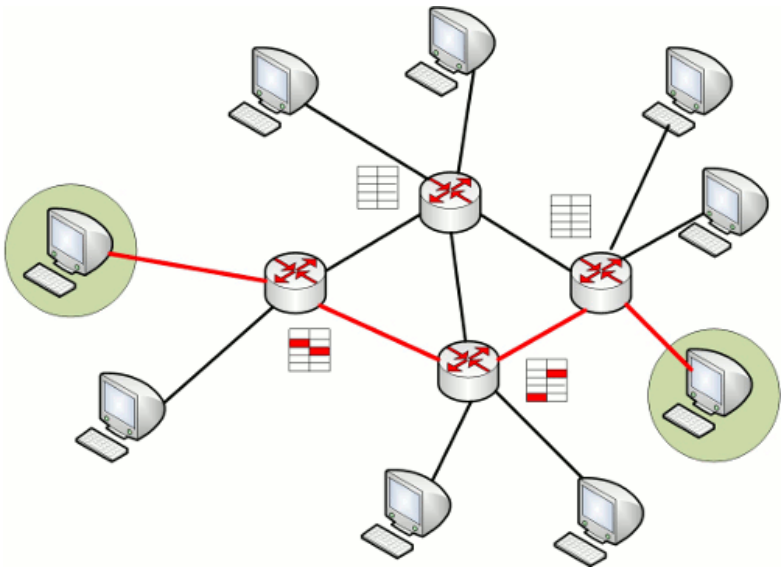


Рисунок 19. Пример работы маршрутизаторов в сети пакетной передачи данных

Важным аспектом работы маршрутизаторов является способ обновления информации в таблицах маршрутизации. Это может выполняться двумя способами вручную и автоматически. В первом случае администратор сети самостоятельно настраивает таблицы маршрутизации. Такой вариант подходит только для небольших сетей, конфигурация которых изменяется редко. Маршрутизаторы первого типа называются статическими. Автоматическое обновление таблиц маршрутизации выполняется с помощью обмена информационными сообщениями между соседними маршрутизаторами о текущей обстановке, а также проверке соединительных каналов между ними. Такие маршрутизаторы называются динамическими. Главный их недостаток заключается в необходимости дополнительных сетевых и вычислительных ресурсов для обмена данными и расчета маршрута. Однако динамические маршрутизаторы могут быть использованы при построении сетей любого масштаба.

Маршрутизаторы бывают как проводные – наиболее классический тип с несколькими портами, в которые подключаются кабели от внешних устройств, так и беспроводные, например, используемые для построения сетей WiFi. Также маршрутизаторы значительно различаются по емкости. Это могут быть как небольшие роутеры с 8-12 портами, которые используются при построении локальных сетей, так и громоздкие модульные конструкции, рассчитанные на сотни подключаемых сегментов.

3.7 Практическое занятие № 7 (2 часа)
Тема: «Протокол TCP/IP»
3.7.1 Задачи работы:
1. Ознакомиться с протоколами TCP/IP
2. Ознакомиться с архитектурой TCP/IP

3.7.2 Краткое описание практического:

Протокол TCP/IP

Архитектура TCP/IP

3.7.3Результаты и выводы:

1. Протоколы TCP/IP.

TCP/IP - это два основных сетевых протокола Internet. Часто это название используют и для обозначения сетей, работающих на их основе. Протокол IP (Internet Protocol - IP v4) обеспечивает маршрутизацию (доставку по адресу) сетевых пакетов. Протокол TCP (Transfer Control Protocol) обеспечивает установление надежного соединения между двумя машинами и собственно передачу данных, контролируя оптимальный размер пакета передаваемых данных и осуществляя перепосылку в случае сбоя. Число одновременно устанавливаемых соединений между абонентами сети не ограничивается, т. е. любая машина может в некоторый промежуток времени обмениваться данными с любым количеством других машин по одной физической линии.

Другое важное преимущество сети с протоколами TCP/IP состоит в том, что по нему могут быть объединены машины с разной архитектурой и разными операционными системами, например Unix, VAX VMS, MacOS, MS-DOS, MS Windows и т.д. Причем машины одной системы при помощи сетевой файловой системы NFS (Net File System) могут подключать к себе диски с файловой системой совсем другой ОС и оперировать "чужими" файлами как своими.

Протоколы TCP/IP (Transmission Control Protocol/Internet Protocol) являются базовыми транспортным и сетевым протоколами в OS UNIX. В заголовке TCP/IP пакета указывается:

IP-адрес отправителяIP-адрес получателяНомер порта (Фактически - номер прикладной программы, которой этот пакет предназначен)

Пакеты TCP/IP имеют уникальную особенность добраться до адресата, пройдя сквозь разнородные в том числе и локальные сети, используя разнообразные физические носители.Маршрутизацию IP-пакета (переброску его в требуемую сеть) осуществляют на добровольных началах компьютеры, входящие в TCP/IP сеть.

Протокол IP - это протокол, описывающий формат пакета данных, передаваемого по сети.

Следующий простой пример можетн прояснить, каким образом происходит передача данных и передача данных. Когда Вы получаете телеграмму, весь текст в ней (и адрес, и сообщение) написан на ленте подряд, но есть правила, позволяющие понять, где тут адрес, а где сообщение. Аналогично, пакет в компьютерной сети представляет собой поток битов, а протокол IP определяет, где адрес и прочая служебная информация, а где

сами передаваемые данные. Таким образом, протокол IP в эталонной модели ISO/OSI является протоколомсетевого (3) уровня.

Протокол TCP - это протокол следующего уровня, предназначеный для контроля передачи и целостности передаваемой информации.

Когда Вы не расслышали, что сказал Вам собеседник в телефонном разговоре, Вы просите его повторить сказанное. Приблизительно этим занимается и протокол TCP применительно к компьютерным сетям. Компьютеры обмениваются пакетами протокола IP, контролируют их передачу по протоколу TCP и, объединяясь в глобальную сеть, образуют Интернет. Протокол TCP является протоколом транспортного (4) уровня.

2. Архитектура TCP/IP.

После того как семейство протоколов TCP/IP было реализовано и внедрено, Международная организация по стандартизации (International Organization for Standardization, ISO) предложила собственную семиуровневую сетевую модель, названную OSI (Open System Interconnection — взаимодействие открытых систем). Она так никогда и не приобрела широкой популярности из-за своей сложности и неэффективности.

В данной модели обмен информацией может быть представлен в виде стека, представленного на рисунке 21. Как видно из рисунка, в этой модели определяется все - от стандарта физического соединения сетей до протоколов обмена прикладного программного обеспечения. Дадим некоторые комментарии к этой модели.

Физический уровень данной модели определяет характеристики физической сети передачи данных, которая используется для межсетевого обмена. Это такие параметры, как: напряжение в сети, сила тока, число контактов на разъемах и т.п. Типичными стандартами этого уровня являются, например RS232C, V35, IEEE 802.3 и т.п.

Рисунок 21. Семиуровневая модель протоколов межсетевого обмена OSI

К канальному уровню отнесены протоколы, определяющие соединение, например, SLIP (Strial Line Internet Protocol), PPP (Point to Point Protocol), NDIS, пакетный протокол, ODI и т.п. В данном случае речь идет о протоколе взаимодействия между драйверами устройств и устройствами, с одной стороны, а с другой стороны, между операционной системой и драйверами устройства. Такое определение основывается на

том, что драйвер - это, фактически, конвертор данных из оного формата в другой, но при этом он может иметь и свой внутренний формат данных.

К сетевому (межсетевому) уровню относятся протоколы, которые отвечают за отправку и получение данных, или, другими словами, за соединение отправителя и получателя. Вообще говоря, эта терминология пошла от сетей коммутации каналов, когда отправитель и получатель действительно соединяются на время работы каналом связи. Применительно к сетям TCP/IP, такая терминология не очень приемлема. К этому уровню в TCP/IP относят протокол IP (Internet Protocol). Именно здесь определяется отправитель и получатель, именно здесь находится необходимая информация для доставки пакета по сети.

Транспортный уровень отвечает за надежность доставки данных, и здесь, проверяя контрольные суммы, принимается решение о сборке сообщения в одно целое. В Internet транспортный уровень представлен двумя протоколами TCP (Transport Control Protocol) и UDP (User Datagramm Protocol). Если предыдущий уровень (сетевой) определяет только правила доставки информации, то транспортный уровень отвечает за целостность доставляемых данных.

Уровень сессии определяет стандарты взаимодействия между собой прикладного программного обеспечения. Это может быть некоторый промежуточный стандарт данных или правила обработки информации. Условно к этому уровню можно отнеси механизм портов протоколов TCP и UDP и Berkeley Sockets. Однако обычно, рамках архитектуры TCP/IP такого подразделения не делают.

Уровень обмена данными с прикладными программами (Presentation Layer) необходим для преобразования данных из промежуточного формата сессии в формат данных приложения. В Internet это преобразование возложено на прикладные программы.

Уровень прикладных программ или приложений определяет протоколы обмена данными этих прикладных программ. В Internet к этому уровню могут быть отнесены такие протоколы, как: FTP, TELNET, HTTP, GOPHER и т.п.

Вообще говоря, стек протоколов TCP отличается от только что рассмотренного стека модели OSI. Обычно его можно представить в виде схемы, представленной на рисунке 22.

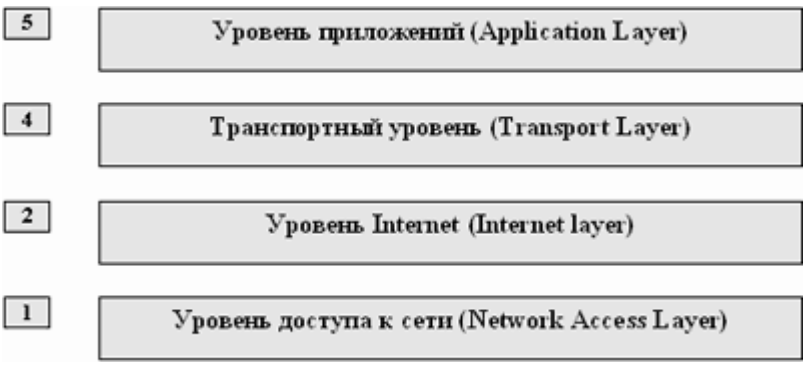


Рисунок 22. Структура стека протоколов TCP/IP

В этой схеме на уровне доступа к сети располагаются все протоколы доступа к физическим устройствам. Выше располагаются протоколы межсетевого обмена IP, ARP, ICMP. Еще выше основные транспортные протоколы TCP и UDP, которые кроме сбора пакетов в сообщения еще и определяют какому приложению необходимо данные отправить или от какого приложения необходимо данные принять. Над транспортным уровнем располагаются протоколы прикладного уровня, которые используются приложениями для обмена данными.

Базируясь на классификации OSI (Open System Integration) всю архитектуру протоколов семейства TCP/IP попробуем сопоставить с эталонной моделью (рисунок 23).

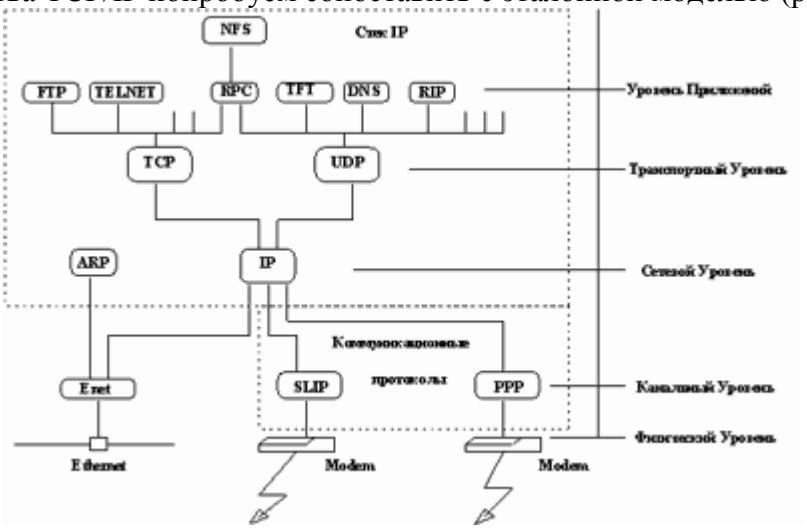


Рисунок 23. Схема модулей, реализующих протоколы семейства TCP/IP в узле сети

Прямоугольниками на схеме обозначены модули, обрабатывающие пакеты, линиями - пути передачи данных. Прежде чем обсуждать эту схему, введем необходимую для этого терминологию.

Драйвер - программа, непосредственно взаимодействующая с сетевым адаптером.

Модуль - это программа, взаимодействующая с драйвером, с сетевыми прикладными программами или с другими модулями.

Схема приведена для случая подключения узла сети через локальную сеть Ethernet, поэтому названия блоков данных будут отражать эту специфику.

Сетевой интерфейс - физическое устройство, подключающее компьютер к сети. В нашем случае - карта Ethernet.

Кадр - это блок данных, который принимает/отправляет сетевой интерфейс.

IP-пакет - это блок данных, которым обменивается модуль IP с сетевым интерфейсом.

UDP-датаграмма - блок данных, которым обменивается модуль IP с модулем UDP.

TCP-сегмент - блок данных, которым обменивается модуль IP с модулем TCP.

Прикладное сообщение - блок данных, которым обмениваются программы сетевых приложений с протоколами транспортного уровня.

Инкапсуляция - способ упаковки данных в формате одного протокола в формат другого протокола. Например, упаковка IP-пакета в кадр Ethernet или TCP-сегмента в IP-пакет. Согласно словарю иностранных слов термин "инкапсуляция" означает "образование капсулы вокруг чужих для организма веществ (инородных тел, паразитов и т.д.)". В рамках межсетевого обмена понятие инкапсуляции имеет несколько более расширенный смысл. Если в случае инкапсуляции IP в Ethernet речь идет действительно о помещении пакета IP в качестве данных Ethernet-фрейма, или, в случае инкапсуляции TCP в IP, помещение TCP-сегмента в качестве данных в IP-пакет, то при передаче данных по коммутируемым каналам происходит дальнейшая "нарезка" пакетов теперь уже на пакеты SLIP или фреймы PPP.

Рисунок 24. Инкапсуляция протоколов верхнего уровня в протоколы TCP/IP

Вся схема (рисунок 24) называется стеком протоколов TCP/IP или просто стеком TCP/IP. Чтобы не возвращаться к названиям протоколов расшифруем аббревиатуры TCP, UDP, ARP, SLIP, PPP, FTP, TELNET, RPC, TFTP, DNS, RIP, NFS:

TCP - Transmission Control Protocol - базовый транспортный протокол, давший название всему семейству протоколов TCP/IP.

UDP - User Datagram Protocol - второй транспортный протокол семейства TCP/IP. Различия между TCP и UDP будут обсуждены позже.

ARP - Address Resolution Protocol - протокол используется для определения соответствия IP-адресов и Ethernet-адресов.

SLIP - Serial Line Internet Protocol (Протокол передачи данных по телефонным линиям).

PPP - Point to Point Protocol (Протокол обмена данными "точка-точка").

FTP - File Transfer Protocol (Протокол обмена файлами).

TELNET - протокол эмуляции виртуального терминала.

RPC - Remote Process Control (Протокол управления удаленными процессами).

TFTP - Trivial File Transfer Protocol (Тривиальный протокол передачи файлов).

DNS - Domain Name System (Система доменных имен).

RIP - Routing Information Protocol (Протокол маршрутизации).

NFS - Network File System (Распределенная файловая система и система сетевой печати).

При работе с такими программами прикладного уровня, как FTP или telnet, образуется стек протоколов с использованием модуля TCP, представленный на рисунке 25.

Рисунок 25. Стек протоколов при использовании модуля TCP

При работе с прикладными программами, использующими транспортный протокол UDP, например, программные средства Network File System (NFS), используется другой стек, где вместо модуля TCP будет использоваться модуль UDP (рисунок 26).

Рисунок 26. Стек протоколов при работе через транспортный протокол UDP

При обслуживании блочных потоков данных модули TCP, UDP и драйвер ENET работают как мультиплексоры, т.е. перенаправляют данные с одного входа на несколько выходов и наоборот, с многих входов на один выход. Так, драйвер ENET может направить кадр либо модулю IP, либо модулю ARP, в зависимости от значения поля "тип" в заголовке кадра. Модуль IP может направить IP-пакет либо модулю TCP, либо модулю UDP, что определяется полем "протокол" в заголовке пакета.

Получатель UDP-датаграммы или TCP-сообщения определяется на основании значения поля "порт" в заголовке датаграммы или сообщения.

Все указанные выше значения прописываются в заголовке сообщения модулями на отправляющем компьютере. Так как схема протоколов - это дерево, то к его корню ведет только один путь, при прохождении которого каждый модуль добавляет свои данные в заголовок блока. Машина, принявшая пакет, осуществляет демультиплексирование в соответствии с этими отметками.

Технология Internet поддерживает разные физические среды, из которых самой распространенной является Ethernet. В последнее время большой интерес вызывает подключение отдельных машин к сети через TCP-стек по коммутируемым (телефонным) каналам. С появлением новых магистральных технологий типа ATM или FrameRelay активно ведутся исследования по инкапсуляции TCP/IP в эти протоколы. На сегодняшний день многие проблемы решены и существует оборудование для организации TCP/IP сетей через эти системы.

3.8 Практическое занятие № 8 (2 часа)
Тема: «Типовая структура процессора»
3.8.1 Задачи работы:
1. Ознакомиться с типовой структурой ВМ на микропроцессорных наборах
2. Ознакомиться с типовой структурой процессора и основной памяти

3.8.2 Краткое описание практического:
Типовая структура ВМ на микропроцессорных наборах
Типовая структура процессора и основной памяти

3.8.3Результаты и выводы:

1. Типовая структура ВМ на микропроцессорных наборах

ППД – память прямого доступа (RAM–randomaccessmemory);
ПЗУ – постоянное запоминающее устройство (ROM–readonlymemory)
АДП – адаптер (контроллер для подключения одного устройства).
ПДП – прямой доступ в память (DMA-direct memory access)

ГТ – генератор тактов (CLG–clockgenerator)
RSG–ResetGenerator
У- шина управления; А- шина адресов; Д- шина данных.
Кроме центрального процессора в состав современной ВМ могут входить сопроцессоры:

- математический сопроцессор (NPR–numericalprocessor);
- аналоговый процессор (APR–analogprocessor);
- процессор цифровой обработки сигналов (ЦОС) (DSP-digitalsignalprocessor).

Раздел 2. Организация процессора и основной памяти ВМ

В разделе идет речь о машинах с контроллерным управлением, в которых порядок выполнения команд явно задается программой. Машины с потоковым управлением и машины с запросным управлением (редукционные) в данном курсе не рассматриваются.

Процессорвыполняет две функции:

- обработка данных в соответствии с заданной программой;
- управление всеми устройствами машины.

Управление в соответствии с заданной программой представляется в виде последовательности команд в цифровой форме. Каждая из команд имеет две части:

Операционная часть	адресная часть

Операционная часть - задает код операции и режим ее выполнения. Адресная часть – содержит сведения о размещении операндов (данных): непосредственно сами значения данных, адреса данных в памяти или сведения для определения адресов размещения данных в памяти. Формирование исполнительного адреса – этап перехода от сведений об адресе к самому адресу. В адресной части могут быть сведения об отсутствии операндов (нульадресная или безадресная команда) и от одного (одноадресная команда) до трех операндов (трехадресная команда).

2. Типовая структура процессора и основной памяти

АЛУ - арифметико-логическое устройство.
РОНы - регистры общего назначения (от 8 до нескольких сотен штук).
ΘРг СС - регистр слова состояния. Содержит текущее состояние процессора, в который входит уровень приоритета текущей программы, биты условий $\{_j\}$завершения последней команды, режим обработки текущей команды. Возможны следующие режимы обработки (в порядке возрастания уровня):

- UserMode- режим пользователя; в этом режиме не могут выполнятся системные команды (команды изменения состояния процессора и команды ввода- вывода);
- SuperVisorMode- режим супервизора; обеспечивается выполнение всех команд ввода-вывода;
- KernelMode- режим ядра; возможно выполнение всех команд процессора.

ПС - программный счетчик. Содержит адрес текущей команды и автоматически наращивается для подготовки адреса следующей команды (исключение составляет команда перехода);
Рг Команд - регистр команд. Содержит код исполняемой в данный момент команды;
ДешКОПиРА – дешифратор кода операции и режимов адресации;
Форм-льУС – формирователь управляющих сигналов $\{ У_i\}$;
РАП - регистр адреса памяти; РДП - регистр данных памяти;
РгУС – регистр управляющего слова контроллера памяти;

3.9 Практическое занятие № 9 (4 часа)

Тема: «Сетевая модель OSI»

3.9.1 Задачи работы:

1. изучить сетевую модель OSI;
2. изучить правила адресации сетевого уровня на примере протокола IP;
3. изучить маршрутизацию IP.

3.9.2 Краткое описание практического:

Сетевая модель OSI

Правила адресации сетевого уровня на примере протокола IP

3.9.3Результаты и выводы:

Модель OSI описывает только системные средства взаимодействия, реализуемые операционной системой, системными утилитами, системными аппаратными средствами. Модель не включает средства взаимодействия приложений конечных пользователей.

Эта модель описывает функции семи иерархических уровней и интерфейсы взаимодействия между уровнями. Каждый уровень определяется сервисом, который он предоставляет вышестоящему уровню, и протоколом - набором правил и форматов данных для взаимодействия между собой объектов одного уровня, работающих на разных компьютерах.

Каждый уровень поддерживает интерфейсы с выше- и нижележащими уровнями. Ниже перечислены (в направлении сверху вниз) уровни модели OSI и указаны их общие функции.

Уровень приложения (Application) - интерфейс с прикладными процессами.

Уровень представления (Presentation) - согласование представления (форматов, кодировок) данных прикладных процессов.

Сеансовый уровень (Session) - установление, поддержка и закрытие логического сеанса связи между удаленными процессами.

Транспортный уровень (Transport) - обеспечение безошибочного сквозного обмена потоками данных между процессами во время сеанса.

Сетевой уровень (Network) - фрагментация и сборка передаваемых транспортным уровнем данных, маршрутизация и продвижение их по сети от компьютера-отправителя к компьютеру-получателю.

Канальный уровень (Data Link) - управление каналом передачи данных, управление доступом к среде передачи, передача данных по каналу, обнаружение ошибок в канале и их коррекция.

Физический уровень (Physical) - физический интерфейс с каналом передачи данных, представление данных в виде физических сигналов и их кодирование.

Принципы выделения этих уровней таковы: каждый уровень отражает надлежащий уровень абстракции и имеет строго определенную функцию. Эта функция выбиралась, прежде всего, так, чтобы можно было определить международный стандарт. Границы уровней выбирались так, чтобы минимизировать поток информации через интерфейсы.

Два самых низших уровня - физический и канальный - реализуются аппаратными и программными средствами, остальные пять более высоких уровней реализуются, как правило, программными средствами (рисунок 1).

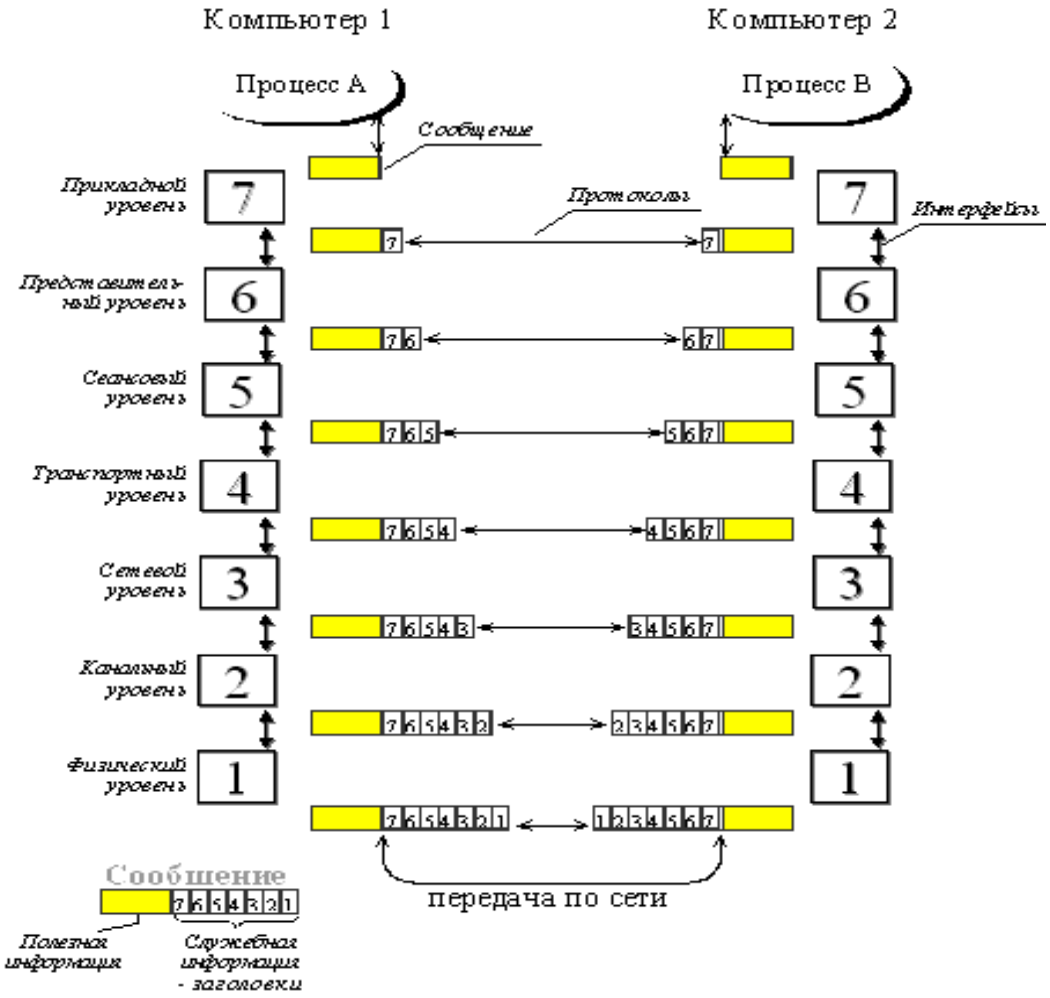


Рисунок 1 - Модель взаимодействия открытых систем ISO/OSI

При продвижении пакета данных по уровням сверху вниз каждый новый уровень добавляет к пакету свою служебную информацию в виде заголовка и, возможно, трейлера (информации, помещаемой в конец сообщения). Эта операция называется инкапсуляцией данных верхнего уровня в пакете нижнего уровня. Служебная информация предназначается для объекта того же уровня на удаленном компьютере, ее формат и интерпретация определяются протоколом данного уровня. Наконец, сообщение достигает нижнего, физического уровня, который, собственно, и передает его по линиям связи машине-адресату. К этому моменту сообщение "обрастает" заголовками всех уровней.

Когда сообщение по сети поступает на другую машину, оно последовательно перемещается вверх с уровня на уровень. Каждый уровень анализирует, обрабатывает и удаляет заголовок своего уровня, выполняет соответствующие данному уровню функции и передает сообщение вышележащему уровню. Тот в свою очередь рассматривает эти данные как пакет со своей служебной информацией и данными для верхнего уровня, и процедура повторяется, пока пользовательские данные, очищенные от всей служебной информации, не достигнут прикладного процесса.

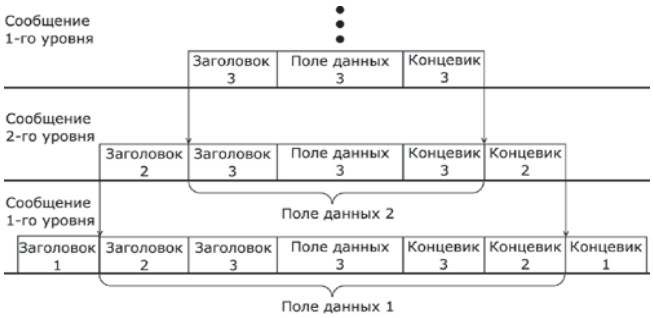


Рисунок 2 – Вложенность сообщений различных уровней

Кроме термина "сообщение" (message) существуют и другие названия, используемые сетевыми специалистами для обозначения единицы обмена данными. В стандартах ISO для протоколов любого уровня используется такой термин как "протокольный блок данных" - Protocol Data Unit (PDU). Кроме этого, часто используются названия кадр (frame), пакет (packet), дейтаграмма (datagram).

Теперь рассмотрим каждый уровень этой модели. Отметим что это модель, а не архитектура сети. Она не определяет протоколов и сервис каждого уровня. Она лишь говорит, что он должен делать.

Физический уровень. Этот уровень имеет дело с передачей битов по физическим каналам, таким, например, как коаксиальный кабель, витая пара или оптоволоконный кабель. К этому уровню имеют отношение характеристики физических сред передачи данных, такие как полоса пропускания, помехозащищенность, волновое сопротивление и другие. На этом же уровне определяются характеристики электрических сигналов, такие как требования к фронтам импульсов, уровням напряжения или тока передаваемого сигнала, тип кодирования, скорость передачи сигналов. Кроме этого, здесь стандартизуются типы разъемов и назначение каждого контакта.

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

В обобщенном виде функции физического уровня заключаются в следующем:

- передача битов по физическим каналам;
- формирование электрических сигналов;
- кодирование информации;
- синхронизация;
- модуляция.

Этот уровень реализуется аппаратно. Примером протокола физического уровня может служить спецификация 10Base-T технологии Ethernet.

Канальный уровень. На физическом уровне просто пересылаются биты. При этом не учитывается, что в некоторых сетях, в которых линии связи используются (разделяются) попеременно несколькими парами взаимодействующих компьютеров, физическая среда передачи может быть занята. Поэтому одной из задач канального уровня является проверка доступности среды передачи. Другой задачей канального уровня является реализация механизмов обнаружения и коррекции ошибок. Для этого на канальном уровне биты группируются в наборы, называемые кадрами (frames). Канальный уровень обеспечивает корректность передачи каждого кадра, помещая специальную последовательность бит в начало и конец каждого кадра, чтобы отметить его, а также вычисляет контрольную сумму, суммируя все байты кадра определенным способом и добавляя контрольную сумму к кадру. Когда кадр приходит, получатель снова вычисляет контрольную сумму полученных данных и сравнивает результат с контрольной суммой из кадра. Если они совпадают, кадр считается правильным и принимается. Если же контрольные суммы не совпадают, то фиксируется ошибка. Необходимо отметить, что

функция исправления ошибок для канального уровня не является обязательной, поэтому в некоторых протоколах этого уровня она отсутствует, например в Ethernet и Frame relay.

Функции канального уровня заключаются в следующем:

- надежная доставка пакета между двумя соседними станциями в сети с произвольной топологией, а также между любыми станциями в сети с типовой топологией;
- проверка доступности разделяемой среды;
- выделение кадров из потока данных, поступающих по сети;
- формирование кадров при отправке данных;
- подсчет и проверка контрольной суммы.

Этот уровень реализуются программно-аппаратно. В протоколах канального уровня, используемых в локальных сетях, заложена определенная структура связей между компьютерами и способы их адресации. Хотя канальный уровень и обеспечивает доставку кадра между любыми двумя узлами локальной сети, он это делает только в сети с совершенно определенной топологией связей, именно той топологией, для которой он был разработан. К таким типовым топологиям, поддерживаемым протоколами канального уровня локальных сетей, относятся общая шина, кольцо и звезда. Примерами протоколов канального уровня являются протоколы Ethernet, Token Ring, FDDI, 100VG-AnyLAN.

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

В глобальных сетях, которые редко обладают регулярной топологией, канальный уровень обеспечивает обмен сообщениями между двумя соседними компьютерами, соединенными индивидуальной линией связи. Примерами протоколов "точка - точка" могут служить широко распространенные протоколы PPP и LAP-B.

Именно так организованы сети X.25. Иногда в глобальных сетях функции канального уровня в чистом виде выделить трудно, так как в одном и том же протоколе они объединяются с функциями сетевого уровня. Примерами такого подхода могут служить протоколы технологий ATM и Frame relay.

В целом канальный уровень представляет собой весьма мощный набор функций по пересылке сообщений между узлами сети.

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

Протоколы канального уровня локальных сетей обеспечивают доставку данных между любыми узлами только в сети с соответствующей типовой топологией, например топологией иерархической звезды. Это жесткое ограничение, которое не позволяет строить сети с развитой структурой, например сети, объединяющие несколько сетей предприятия в единую сеть, или высоконадежные сети, в которых существуют избыточные связи между узлами.

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

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

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

Свойства сетевого уровня — доставка пакета:

- между любыми двумя узлами сети с произвольной топологией;
- между любыми двумя сетями в составной сети.

При этом, сеть — это совокупность компьютеров, использующих для обмена данными единую сетевую технологию; а маршрут — последовательность прохождения пакетом маршрутизаторов в составной сети.

На рисунке 3 показаны четыре сети, связанные тремя маршрутизаторами. Между узлами A и B данной сети пролегает два маршрута: первый — через маршрутизаторы 1 и 3, а второй — через маршрутизаторы 1, 2 и 3.

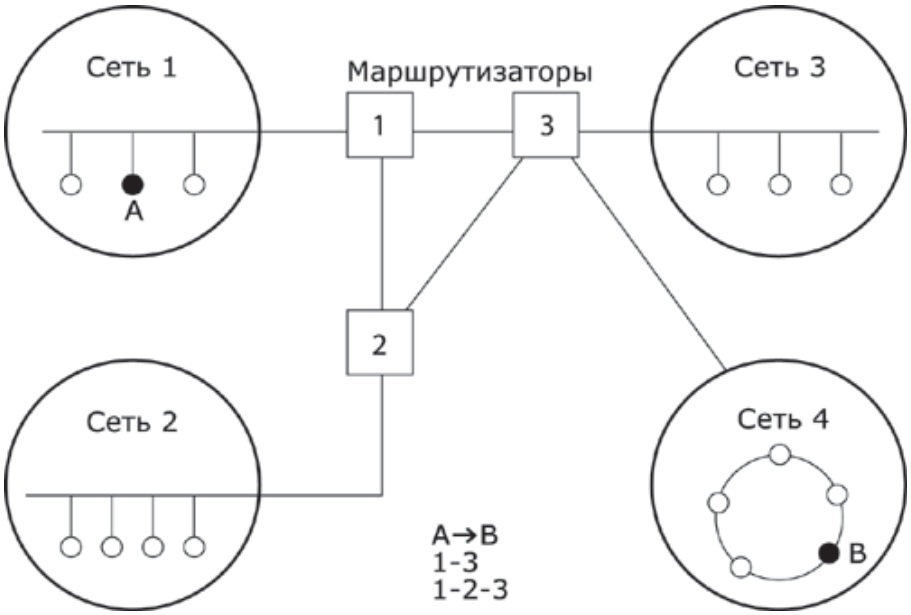


Рисунок 3 - Пример составной сети

В общем случае функции сетевого уровня шире, чем функции передачи сообщений по связям с нестандартной структурой, которые мы рассмотрели на примере объединения нескольких локальных сетей. Сетевой уровень также решает задачи согласования разных технологий, упрощения адресации в крупных сетях и создания надежных и гибких барьеров на пути нежелательного трафика между сетями.

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

На сетевом уровне определяется два вида протоколов. Первый вид — сетевые протоколы (routed protocols) — реализуют продвижение пакетов через сеть. Именно эти

протоколы обычно имеют в виду, когда говорят о протоколах сетевого уровня. Однако часто к сетевому уровню относят и другой вид протоколов, называемых протоколами обмена маршрутной информацией или просто протоколами маршрутизации (routing protocols). С помощью этих протоколов маршрутизаторы собирают информацию о топологии межсетевых соединений. Протоколы сетевого уровня реализуются программными модулями операционной системы, а также программными и аппаратными средствами маршрутизаторов.

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

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

Транспортный уровень. На пути от отправителя к получателю пакеты могут быть искажены или утеряны. Хотя некоторые приложения имеют собственные средства обработки ошибок, существуют и такие, которые предпочитают сразу иметь дело с надежным соединением. Работа транспортного уровня заключается в том, чтобы обеспечить приложениям или верхним уровням стека - прикладному и сеансовому - передачу данных с той степенью надежности, которая им требуется. Модель OSI определяет пять классов сервиса, предоставляемых транспортным уровнем. Эти виды сервиса отличаются качеством предоставляемых услуг: срочностью, возможностью восстановления прерванной связи, наличием средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол, а главное - способностью к обнаружению и исправлению ошибок передачи, таких как искажение, потеря и дублирование пакетов.

Выбор класса сервиса транспортного уровня определяется, с одной стороны, тем, в какой степени задача обеспечения надежности решается самими приложениями и протоколами более высоких, чем транспортный, уровней, а с другой стороны, этот выбор зависит от того, насколько надежной является вся система транспортировки данных в сети. Так, например, если качество каналов передачи связи очень высокое, и вероятность возникновения ошибок, не обнаруженных протоколами более низких уровней, невелика, то разумно воспользоваться одним из облегченных сервисов транспортного уровня. Если же транспортные средства изначально очень ненадежны, то целесообразно обратиться к наиболее развитому сервису транспортного уровня, который работает, используя максимум средств для обнаружения и устранения ошибок - с помощью предварительного установления логического соединения, контроля доставки сообщений с помощью контрольных сумм и циклической нумерации пакетов, установления тайм-аутов доставки и т.п.

Т.о. функции транспортного уровня - это обеспечение доставки информации с требуемым качеством между любыми узлами сети:

- разбивка сообщения сеансового уровня на пакеты, их нумерация;
- буферизация принимаемых пакетов;
- упорядочивание прибывающих пакетов;
- адресация прикладных процессов;
- управление потоком.

Как правило, все протоколы, начиная с транспортного уровня и выше, реализуются программными средствами конечных узлов сети - компонентами их сетевых операционных систем. В качестве примера транспортных протоколов можно привести протоколы TCP и UDP стека TCP/IP и протокол SPX стека Novell.

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

Сеансовый уровень. Сеансовый уровень (Session layer) обеспечивает управление диалогом: фиксирует, какая из сторон является активной в настоящий момент, предоставляет средства синхронизации. Последние позволяют вставлять контрольные точки в длинные передачи, чтобы в случае отказа можно было вернуться назад к последней контрольной точке, а не начинать все сначала. На практике немногие приложения используют сеансовый уровень, и он редко реализуется в виде отдельных протоколов, хотя функции этого уровня часто объединяют с функциями прикладного уровня и реализуют в одном протоколе.

Функции сеансового уровня - это управление диалогом объектов прикладного уровня:

- установление способа обмена сообщениями (дуплексный или полудуплексный);
- синхронизация обмена сообщениями;
- организация "контрольных точек" диалога.

Уровень представления. Этот уровень обеспечивает гарантию того, что информация, передаваемая прикладным уровнем, будет понятна прикладному уровню в другой системе. При необходимости уровень представления выполняет преобразование форматов данных в некоторый общий формат представления, а на приеме, соответственно, выполняет обратное преобразование. Таким образом, прикладные уровни могут преодолеть, например, синтаксические различия в представлении данных. На этом уровне может выполняться шифрование и дешифрование данных, благодаря которому секретность обмена данными обеспечивается сразу для всех прикладных сервисов. Примером протокола, работающего на уровне представления, является протокол Secure Socket Layer (SSL), который обеспечивает секретный обмен сообщениями для протоколов прикладного уровня стека TCP/IP.

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

- преобразование данных из внешнего формата во внутренний;
- шифрование и расшифровка данных.

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

Существует очень большое разнообразие протоколов прикладного уровня. Среди них хотелось бы отметить такие как FTP, Telnet, HTTP.

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

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

- идентификация, проверка прав доступа;
- принт- и файл-сервис, почта, удаленный доступ.

Три нижних уровня - физический, канальный и сетевой - являются сетезависимыми, то есть протоколы этих уровней тесно связаны с технической реализацией сети, с используемым коммуникационным оборудованием.

Три верхних уровня - сеансовый, уровень представления и прикладной - ориентированы на приложения и мало зависят от технических особенностей построения сети. На протоколы этих уровней не влияют никакие изменения в топологии сети, замена оборудования или переход на другую сетевую технологию.

Транспортный уровень является промежуточным, он скрывает все детали функционирования нижних уровней от верхних уровней. Это позволяет разрабатывать приложения, независящие от технических средств, непосредственно занимающихся транспортировкой сообщений.

Рисунок 4 показывает уровни модели OSI, на которых работают различные элементы сети. Компьютер, с установленной на нем сетевой ОС, взаимодействует с другим компьютером с помощью протоколов всех семи уровней. Это взаимодействие компьютеры осуществляют через различные коммуникационные устройства. В зависимости от типа, коммуникационное устройство может работать либо только на физическом уровне (повторитель), либо на физическом и канальном (мост и коммутатор), либо на физическом, канальном и сетевом, иногда захватывая и транспортный уровень (маршрутизатор).

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

Поэтому модель OSI стоит рассматривать, в основном, как опорную базу для классификации и сопоставления протокольных стеков.

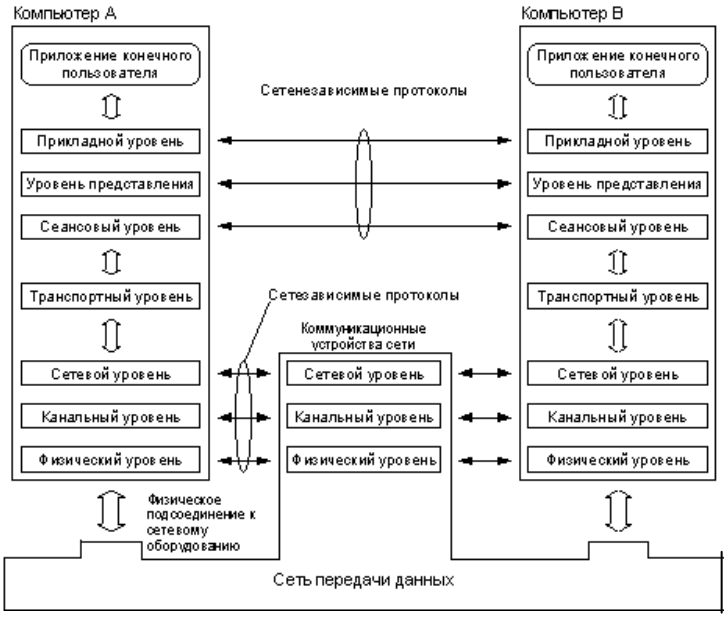


Рисунок 4 - Сетезависимые и сетенезависимые уровни модели OSI

Протокол IP

Архитектуру сетевого уровня удобно рассматривать на примере сетевого протокола IP – самого распространенного в настоящее время, основного протокола сети Интернет. Термин «стек протоколов TCP/IP» означает «набор протоколов, связанных с IP и TCP(протоколом транспортного уровня)».

Архитектура протоколов TCP/IP предназначена для объединенной сети, состоящей из соединенных друг с другом шлюзами отдельных разнородных пакетных подсетей, к которым подключаются разнородные машины.

Каждая из подсетей работает в соответствии со своими специфическими

требованиями и имеет свою природу средств связи. Однако предполагается, что каждая подсеть может принять пакет информации (данные с соответствующим сетевым заголовком) и доставить его по указанному адресу в этой конкретной подсети.

Не требуется, чтобы подсеть гарантировала обязательную доставку пакетов и имела надежный сквозной протокол.

Таким образом, две машины, подключенные *к одной подсети*, могут обмениваться пакетами.

Когда необходимо передать пакет между машинами, *подключенными к разным подсетям*, то машина-отправитель посылает пакет в соответствующий шлюз (шлюз подключен к подсети также как обычный узел). Оттуда пакет направляется по определенному маршруту через систему шлюзов и подсетей, пока не достигнет шлюза, подключенного к той же подсети, что и машина-получатель: там пакет направляется к получателю.

Таким образом, адрес получателя должен содержать в себе:

- номер (адрес) подсети;
- номер (адрес) участника (хоста) внутри подсети.

IP адреса представляют собой 32-х разрядные двоичные числа. Для удобства их записывают в виде четырех десятичных чисел, разделенных точками. Каждое число является десятичным эквивалентом соответствующего байта адреса (для удобства будем записывать точки и в двоичном изображении).

Двоичное	Десятичное
10000000	**128**
11000000	**192**
11100000	**224**
11110000	**240**
11111000	**248**
11111100	**252**
11111110	**254**
11111111	**255**

192.168.200.47 является десятичным эквивалентом двоичного адреса

11000000.10101000.11001000.00101111

Иногда применяют десятичное значение IP-адреса. Его легко вычислить

$$192*256^3+168*256^2+200*256+47=3232286767$$

или с помощью метода Горнера :

$$(((192*256)+168)*256+200)*256+47=3232286767$$

Количество разрядов *адреса подсети* может быть различным и определяется *маской сети*.

Маска сети также является 32-х разрядным двоичным числом. Разряды маски имеют следующий смысл: если разряд маски равен 1, то соответствующий разряд адреса является разрядом адреса подсети, а если 0, то разрядом хоста внутри подсети. Все единичные разряды маски (если они есть) находятся в старшей (левой) части маски, а нулевые (если они есть) – в правой (младшей).

Исходя из вышесказанного, маску часто записывают в виде числа единиц в ней содержащихся.

255.255.248.0 (11111111.11111111.11111000.00000000) – является правильной маской

подсети **(/21)**, а **255.255.250.0 (11111111.11111111.1111<u>101</u>0.00000000)** – является <u>неправильной</u>**,** недопустимой.

Нетрудно увидеть, что *максимальный размер подсети* может быть только степенью двойки (двойку надо возвести в степень, равную количеству нулей в маске).

При передаче пакетов используются *правила маршрутизации*, главное из которых звучит так: «Пакеты участникам своей подсети доставляются напрямую, а остальным – по другим правилам маршрутизации».

Таким образом, требуется определить, является ли получатель членом нашей подсети или нет.

1. ***Определение диапазона адресов подсети.***

Определение диапазона адресов подсети можно произвести из определения понятия маски:

В те разряды, которые относятся к адресу подсети, у всех хостов подсети должны быть *одинаковы*;

В адреса хостов в подсети могут быть *любыми*.

То есть, если наш адрес **192.168.200.47** и маска равна **/20,** то диапазон можно посчитать:

11000000.10101000.11001000.00101111 –адрес
<u>11111111.11111111.11110000.00000000</u> –маска
11000000.10101000.1100XXXX.XXXXXXXX –диапазон адресов

где 0,1 – определенные значения разрядов, X – любое значение,

Что приводит к диапазону адресов:

от **11000000.10101000.11000000.00000000 (192.168.192.0)** до **11000000.10101000.11001111.11111111 (192.168.207.255)**

Следует учитывать, что некоторые адреса являются запрещенными или служебными и их нельзя использовать для адресов хостов или подсетей. Этоадреса, содержащие:

3. **0** в *первом* или *последнем* байте,
4. **255** в *любом* байте (это широковещательные адреса),
5. **127** в *первом* байте (внутренняя петля – этот адрес имеется в каждом хосте и служит для связывания компонентов сетевого уровня). Поэтому доступный диапазон адресов будет несколько меньше.

3.10 Практическое занятие № 10 (4 часа)
Тема: «Запоминающие устройства ЭВМ»
3.10.1 Задачи работы:
1. Ознакомиться с запоминающим устройством ЭВМ
3.10.2 Краткое описание практического:
Запоминающие устройство ЭВМ
3.10.3Результаты и выводы:

Запоминающие устройства (ЗУ) предназначены для хранения информации, а именно - программ и данных.
ЗУ подразделяются на:
- сверхоперативные;
- оперативные;
- внешние.
Оперативной память - это основная память машины. В соответствии с принципом фон Неймана она предназначена для хранения программы (программ в многозадачном режиме), выполняемой ЭВМ в данный момент времени и необходимых для нее данных.
Назначение внешнего ЗУ - хранение массивов информации.
Сверхоперативное ЗУ (СОЗУ) обеспечивает увеличение быстродействия ЭВМ, благодаря использованию команд меньшей (сокращенной) длины и более быстрой элементной базы ее регистров.
Рисунок 1.1- Иерархическая структура ЗУ
Внешние запоминающие устройства (ВЗУ) чаще всего выполняются на барабанах, магнитных и оптических дисках, ленточных накопителях.
Первые два типа ВЗУ называют устройствами прямого доступа (циклического доступа). Магнитные и оптические поверхности этих устройств непрерывно вращаются, чем обеспечивается быстрый доступ к хранимой информации (время доступа этих устройств составляет от нескольких мс до десятка мс). Накопители на магнитных лентах (МЛ) называют устройствами последовательного доступа, из-за последовательного просмотра участков носителя информации (время доступа этих устройств составляет от нескольких секунд до нескольких минут).
Оперативная память делится на:
- оперативные ЗУ (ОЗУ) (Оперативные ЗУ с произвольной выборкой ЗУПВ);
- постоянные ЗУ (ПЗУ).
Назначение и функции оперативных ЗУ: запись и чтение информации.
Назначение и функции ПЗУ - только чтение информации.
Выполняются на ферритовых сердечниках, полупроводниковых микросхемах, магнитных доменах (тонких пленках) и др. Время обращения к памяти составляет от нескольких нс до нескольких мкс.
Сверхоперативная память выполняется на элементной базе процессора, входит в структуру процессора и позволяет значительно повысить его производительность.

Основные характеристики ЗУ

Основная характеристика ЗУ (любого типа) - емкость памяти. Определяет максимальное количество информации, которое может в ней храниться. Емкость может измеряться в битах, байтах или машинных словах. Наиболее распространенной единицей измерения является байт. При большом размере памяти ее емкость выражают в килобайтах (Кбайт) - 1024 байт, в мегобайтах (Мбайт) - миллион байт (точнее 1024*1024 байт), в гигобайтах (Гбайт) - миллиард байт.
Время обращения к памяти. Время обращения при чтении:
, где
$\mathbf{t_д}$ - время доступа (подготовительное время) - промежуток времени между началом операции обращения и моментом начала процесса чтения;
$t_{чт}$ - продолжительность физического процесса считывания;
$t_{рег}$ - время регенерации (восстановления), если в процессе чтения информации произошло ее разрушение.
Время обращения при записи:
, где
$\mathbf{t_п}$ - время подготовки, расходуемое на приведение запоминающих элементов в исходном состоянии, если это необходимо;

$t_{зп}$ - время, необходимое для физического изменения состояния запоминающих элементов при записи информации.

Цикл памяти. Принимается равным минимальному допустимому интервалу между двумя обращениями в память:

.

Положим, что процессы чтения и записи имеют следующие временные диаграммы:

Структура ОЗУ с произвольной выборкой (ЗУПВ)

В оперативных ЗУ с произвольной выборкой (ЗУПВ) запись или чтение в/из памяти осуществляется по адресу, указанному регистром адреса (РА). Чтение или запись слова осуществляется за один цикл. Информация, необходимая для осуществления процесса записи - чтения поступает из процессора, а именно: адрес, данные и управляющие сигналы.

Адресная часть с процессора сначала поступает на регистр адреса (РА), а с него - на дешифратор адреса ДшА, который выбирает строку запоминающего массива (номер ячейки памяти). По сигналу запись (Зп) производится запись данных в заданную ячейку памяти.

Запоминающий массив содержит множество одинаковых запоминающих элементов В памяти статического типа в их качестве используются электронные триггеры, в динамической памяти - полевые транзисторы, работающие на принципе накопления заряда в области затвор-исток.

Структура микросхем динамической памяти (DRAM) в целом близка к структуре статической памяти. Для уменьшения количества выводов (а следовательно, габаритов и стоимости), в микросхемах динамической памяти (DRAM) используется мультиплексированная ША. Полное количество разрядов ША, подаваемое на микросхему DRAM делится на две части - адрес строки и адрес столбца. При адресации ячеек DRAM эти части адреса, последовательно во времени, подаются на адресные входы микросхемы в сопровождении соответственно стробов адреса строки (RAS) и столбца (CAS).

Разделение полного адреса запоминающего элемента и последовательную выдачу его на микросхему осуществляет мультиплексор, являющийся частью контроллера динамической памяти.

Матрица элементов памяти (МЭП) микросхемы DRAM разбита на строки, количество которых равно 2^n, где n - количество разрядов адреса строки или столбца. При вводе адреса строки выбранная строка МЭП считывается в регистр-защелку статического типа, входящего в состав микросхемы DRAM. При считывании строки ее содержимое разрушается, но копия содержимого строки оказывается записанной в регистр - защелку.

Подача адреса столбца в сопровождении строба CAS выбирает в регистре - защелке, в зависимости от организации микросхемы DRAM, бит, тетраду, байт и т.д. При появлении сигнала чтения выбранная информация выдается на ШД после чего записывается на прежнее место в строку МЭП.

При записи информация, поступившая на микросхему DRAM с ШД, записывается сначала в соответствующие разряды регистра - защелки, после чего его содержимое переписывается в прежнюю строку микросхемы DRAM.

ОЗУ магазинного типа (стековая память)

Магазинная (стековая) память организуется по принципу «последний пришел, первый вышел» (LIFO), или «первый пришел, первый вышел» (FIFO). Такая организация памяти является фактически безадресной. Однако регистр адреса в такой памяти имеется и называется указателем стека (УС) (SP-Steak Pointer).

В первом типе памяти новое слово заносится в верхнюю ячейку, ранее занесенные данные проталкиваются вниз. При считывании наоборот, последнее слово выталкивается вверх первым.

В случае организации типа FIFO новое слово заносится в верхнюю ячейку, ранее записанные слова выталкиваются вниз.

Перед началом работы в указатель стека занося начальный адрес. Дальнейшая адресация осуществляется автоматически при выполнении операции записи - чтения путем увеличения - уменьшения адреса на единицу. Физический процесс записи и считывания данных происходит точно так же, как в обычной памяти с произвольным обращением.

Если число слов, записанных в стек и считанных из стека равно, то стек приходит в исходное состояние.

Ассоциативные ЗУ

Выборка информации в ассоциативных ЗУ (АЗУ) осуществляется по содержанию. Пусть, например, в АЗУ хранятся данные на студентов: год рождения, место работы, стаж работы, № группы и т.д., при этом необходимо выбрать фамилии всех студентов 1980 года рождения и работающих. Эта информация является ключом для поиска. Ассоциативная память позволяет выбрать эту информацию за один или несколько циклов памяти. Обычные (программные) средства поиска и сортировки будут работать очень долго.

Каждая ячейка такого АЗУ должна содержать (см. рисунок 1.6.1) регистр для хранения слова данных (как в обычных ОЗУ) и специальные комбинационные логические схемы для сравнения текущего содержимого регистра с ключевым словом, поступающим одновременно на вход всех ячеек. При поиске формируется сигнал чтения из всех ячеек АЗУ. Импульс опроса появится на выходе ячейки, в которой содержимое совпадает с ключом.

Перед началом работы информация заносится в регистр, называемый компарандом (операнд в операции сравнения). Каждая ячейка связана с процессором через регистр признаков (Рг. Пр) с помощью разряда Tj. Регистр признака называют памятью отклика. Регистр маски маскирует те разряды компаранда, которые не должны участвовать в операции сравнения.

Перед началом работы все разряды Рг. Пр устанавливаются в состояние «Лог. 0». По команде процессора «Сравнить» любая ячейка, содержащая слово, которое совпадает с компарандом, формирует сигнал, устанавливающий соответствующий разряд в Рг. Пр в состояние «Лог. 1». Эта информация является адресной для линейной выборки.

3.11 Практическое занятие № 11 (4 часа)

Тема: «Протоколы и алгоритмы маршрутизации»

3.11.1 Задачи работы:

1. Научиться работать с таблицей маршрутизацией.

3.11.2 Краткое описание практического:

Протоколы маршрутизации

Алгоритмы маршрутизации

3.11.3Результаты и выводы:

В сетях, основанных на протоколе IP, *концепция маршрутизации* является одной из важных. Она создает или разбивает сеть. Неправильная конфигурация маршрутизации способна вывести из строя сеть.

Маршрутизация – технология определения пути доставки (маршрута) пакетов. Основные принципы маршрутизации:

4. Каждая операционная система, поддерживающая стек **TCP/IP**, имеет маршрутизатор и таблицу маршрутизации.

5. Таблица маршрутизации используется только тогда, когда определяется, как доставлять пакеты.
6. Маршрутизация должна быть сконфигурирована корректно на обоих концах связи и на каждом участке между ними.

Для определения пути доставки пакета используется *таблица маршрутизации*. Пример таблицы маршрутизации можно получить командой **route** с параметром *print*.

Активные маршруты:

Сетевой адрес	Маска сети	Адрес шлюза	Интерфейс	Метрика
Network Destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.4.1	192.168.4.7	1
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.4.0	255.255.255.0	192.168.4.7	192.168.4.7	1
192.168.4.7	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.4.255	255.255.255.255	192.168.4.7	192.168.4.7	1
224.0.0.0	224.0.0.0	192.168.4.7	192.168.4.7	1
255.255.255.255	255.255.255.255	192.168.4.7	192.168.4.7	1
Основной шлюз:	192.168.1.1			

Рисунок 1. Пример таблицы маршрутизации

В общем случае для маршрутизации используется следующий *алгоритм*. Из пакета извлекается IP-адрес назначения пакета и производится попытка сопоставить его с адресом назначения (*Сетевой адрес*) каждого элемента таблицы маршрутизации пока не найдется наилучшее совпадение. Если совпадений не найдено, то пакет удаляется и отправителю пакета может отправиться сообщение об ошибке. Сравнение производится с тремя порциями информации: **Сетевой адрес (*Network Destination*), Маска сети (*Netmask*) и IP-адрес назначения пакета.**

В основном, производится побитная операция **AND** между **IP-адресом получателя** и **Маской сетии (*Netmask*)**: если полученное значение равно **Сетевому адресу (*Network Destination*)**, то считается, что совпадение найдено.

Пример 1. Необходимо проверить почту на сервере, чей адрес **192.168.4.100** (используется таблица маршрутизации приведенная ранее). Необходимо выполнить побитную операцию **AND** над **IP-адресом получателя пакетов** и **сетевыми масками (*Netmask*)** из таблицы маршрутизации. Эта операция производится над всем масками из таблицы маршрутизации. Но в рассматриваемом примере только 3-я строка наиболее походит.

	1-й октет								2-й октет								3-й октет								4-й октет							
биты	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ID-сети																													ID-узла		
IP-адрес	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1	0	0
десятичная запись	192								168								4								100							
Маска	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
десятичная запись	255								255								255								0							
Результат операции AND	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
десятичная запись	192								168								4								0							

Рисунок 2. Пример определения маршрута доставки пакетов

Как видно из приведенной таблицы, результат побитной операции **AND** совпадает с 3-й строкой таблицы маршрутизации (Рисунок 2). Следовательно, пакет отправится по указанному маршруту через интерфейс**192.168.4.7**.

Следует отметить, что указанный в примере IP-адрес после выполнения побитной операции **AND** над масками совпадет больше чем с одной строкой маршрутизации. Для избежания таких случаев используется*приоритет маршрутов*. Система ищет более точное совпадение адреса с маской (255.255.255.255 более точна, чем 255.255.255.0, которая в свою очередь, более точна, чем 0.0.0.0). Маршрут с сетевым адресом 0.0.0.0 и маской 0.0.0.0 является *маршрутом по умолчанию*. Так как этот маршрут подходит к любому адресу назначения, он описывает маршрут, который используется, если не найден более подходящий. Обычно этот маршрут используется для пересылки пакетов провайдеру Интернет-услуг, при подключении к Интернету.

Для работы с таблицей маршрутизации используется стандартная утилита **ROUTE,** которая выводит на экран и изменяет записи в локальной таблице IP-маршрутизации.

Запущенная без параметров, команда **route** выводит справку.

Параметр	**Описание**
add	Добавление маршрута
change	Изменение существующего маршрута
delete	Удаление маршрута или маршрутов
print	Печать маршрута или маршрутов

Таблица 1. Назначение параметров команды ***route***

Пример 2. Добавление маршрута.

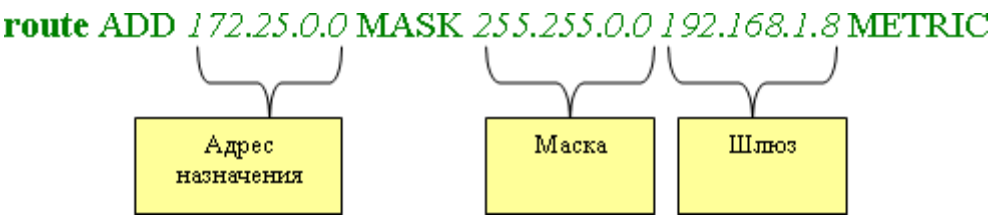


Рисунок 3. СТрока для добавление маршрута

Задание 1. Создайте таблицу для облегчения определения маршрутов.

9. Откройте **табличный процессор** и сформируйте таблицу по следующему шаблону:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG
1		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
2	**IP-адрес**	192								168								252								56							
3	двоичная запись	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0
4	**Маска**	255								255								255								7							
5	двоичная запись	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1
6	**Операция AND**	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
7	**Адрес назначения**																																

Рисунок 4. Образец оформления таблицы

10. Введите в диапазон ячеек **Z3:AG3** формулы для перевода числа в десятичной системе счисления из ячейки **Z2** в двоичную форму (в соответствии с таблицей).

Таблица 2. Формулы для перевода в двоичную систему счисления

Имя Ячейки	**Формула**
AG3	=Z2-2*INT(Z2/2)
AF3	=INT(Z2/2)-2*INT(INT(Z2/2)/2)
AE3	=INT(INT(Z2/2)/2)-2*INT(INT(INT(Z2/2)/2)/2)
AD3	=INT(INT(INT(Z2/2)/2)/2)-2*INT(INT(INT(INT(Z2/2)/2)/2)/2)
AC3	=INT(INT(INT(INT(Z2/2)/2)/2)/2)- 2*INT(INT(INT(INT(INT(Z2/2)/2)/2)/2)/2)
AB3	=INT(INT(INT(INT(INT(Z2/2)/2)/2)/2)/2)- 2*INT(INT(INT(INT(INT(INT(Z2/2)/2)/2)/2)/2)/2)
AA3	=INT(INT(INT(INT(INT(INT(Z2/2)/2)/2)/2)/2)/2)- 2*INT(INT(INT(INT(INT(INT(INT(Z2/2)/2)/2)/2)/2)/2)/2)
Z3	=INT(INT(INT(INT(INT(INT(INT(Z2/2)/2)/2)/2)/2)/2)/2)

11. Аналогично введите формулы для преобразования чисел из десятичной системы счисления в двоичную для ячеек **R2,J2,B2**.
12. Аналогично введите формулы для преобразования маски подсети в двоичную систему счисления.
13. Введите формулы для побитной операции **AND** над **IP-адресом** и **маской (*Netmask*)**:
    - введите в ячейку **AG6** формулу **=AND(AG3;AG5)**;
    - скопируйте введенную формулу в диапазон ячеек **B6:AF6**.
14. Введите в ячейку **Z7** формулу для преобразования 4-го октета маски в десятичную систему счисления - **=AG6*2^AL1+AF6*2^AF1+AE6*2^AE1+AD6*2^AD1+AC6*2^AC1+AB6*2^AB1+AA6*2^AA1+Z6*2^Z1**.
15. Аналогично введите формулы для ячеек **R7, J7, B8**.
16. Сохраните файл в своем каталоге с именем ***ROUTE***.

Задание 2. Создайте новый маршрут для вашего компьютера и проследите его.

9. Запустите виртуальную машину **VM-1** и загрузите ОС **Windows**.
10. Откройте **консоль (*Пуск/Программы/Стандартные/Командная строка*)**.
11. Определите IP-адрес вашего компьютера с помощью утилиты **ipconfig**.
12. Просмотрите таблицу маршрутизации на вашем компьютере:

- выведите справку по команде **route** (для этого необходимо ввести команду и нажать клавишу ***ENTER***);

```
route
```

- выведите таблицу маршрутизации командой **route** с параметром *PRINT*:

```
route PRINT
```

- запомните маршрут по умолчанию (первая строка).

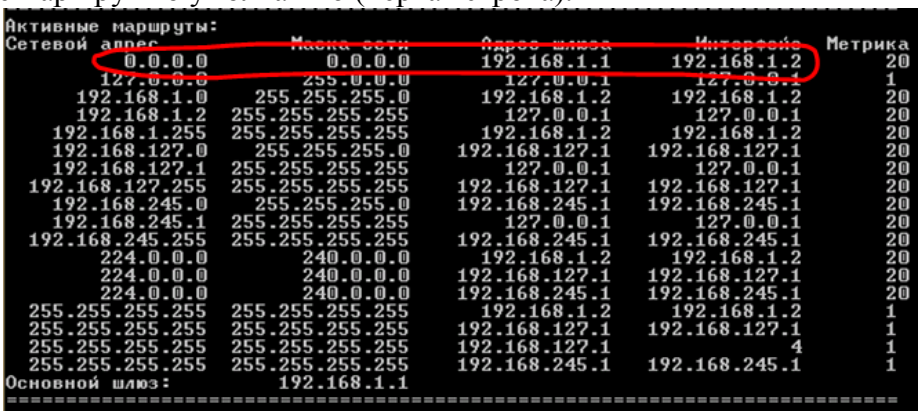


Рисунок 5.Пример вывода программы **ROUTE**

13. Проследите работу маршрутизатора с помощью утилиты *TRACERT*, отправив пакеты на узел **www.opennet.ru**. Введите:

    tracert www.opennet.ru

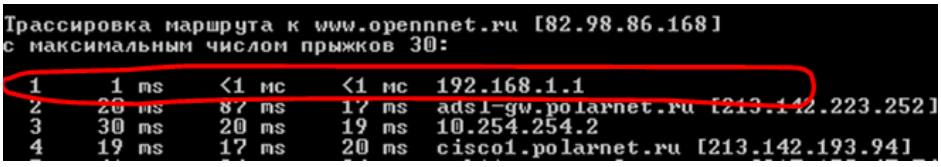


Рисунок 6. Пример вывода программы **TRACERT**

14. *Следует отметить, что пакеты на указанный сайт отправляются через один шлюз (**192.168.1.1**), который видно в первых строках вывода программ **ROUTE** и **TRACERT**.*
15. Добавьте в таблицу маршрутизации компьютера строку для пересылки пакетов в сеть **172.21.0.0** (маска **255.255.0.0**) через сетевой интерфейс компьютера. Введите:

```
route add 172.21.0.0 mask 255.255.0.0 192.168.1.4 METRIC 3
```

16. Проверьте работу внесенных вами изменений с помощью утилиты *TRACERT*.

3.12 Практическое занятие № 12 (4 часа)
Тема: «Устройства ввода вывода информации в ЭВМ»
3.12.1 Задачи работы:
1. Научиться работать с клавиатурой

2. Научиться работать с мышкой и другими манипуляторами
3. Научиться работать со сканером
4. Научиться вводить информацию с бумажных носителей
5. Научиться работать с цифровой камерой
6. Научиться работать с дигитайзером

3.12.2 Краткое описание практического:

Клавиатура
Мышка и другие манипуляторы
Принципы ввода информации с бумажных носителей
Сканер
Цифровая камера
Дигитайзер

3.12.3Результаты и выводы:

1. Клавиатура

Трудно сказать, может ли существовать более важное и универсальное устройство ввода информации в компьютер, чем клавиатура. Вполне возможно, в скором будущем, когда человек будет общаться со своим компьютером посредством жестов, мимики, графических образов, видеоизображений и речи, клавиатуру потеснят другие средства ввода информации. Однако сегодня, когда текст и символы как носители ценной информации еще столь важны, клавиатура обязательно входит в конфигурацию поставляемых персональных компьютеров. Компьютер без клавиатуры - это неполноценный компьютер!

По расположению клавиш настольные клавиатуры делятся на два основных типа, функционально ничуть не уступающие друг другу. В первом варианте функциональные клавиши располагаются в двух вертикальных рядах, а отдельных группы клавиш управления курсором нет. Всего в такой клавиатуре 84 клавиши. Этот стандарт используется в персоналках типа IBM PC, XT и AT до конца 80-х годов. Поэтому некоторые считают этот стандарт устаревшим. Однако многие профессионалы все еще предпочитают именно такую клавиатуру. Между порочим, большинство компьютеров средней и большой мощности по сей день комплектуются именно такой "устаревшей" клавиатурой.

Второй вариант клавиатуры, которую принято называть усовершенствованной, имеет 101 или 102 клавиши. Клавиатурой такого типа снабжаются сегодня почти все настольные персональные компьютеры. Профессионалы не любят эту клавиатуру из-за того, что к функциональным клавишам приходиться далеко тянуться, в самый верхний ряд клавиш, через всю буквенную клавиатуру. Однако количество функциональных клавиш в усовершенствованной клавиатуре не 10, а все 12. Да и другие дополнительные удобства и усовершенствования нравятся многим пользователям. Логично выделены группы клавиш для работы с текстами и управления курсором, продублированы некоторые специальные клавиши, позволяющие более эргономично работать обеими руками. Впрочем, какая клавиатура удобнее - каждый должен решать сам. Ведь поменять клавиатуру в настольном компьютере совсем нетрудно.

Другое дело портативный компьютер, в котором клавиатура обычно является встроенной частью конструкции. Клавиатуры портативных компьютеров в той или иной степени похожи на оба типа клавиатур настольных компьютеров, хотя из-за недостатка места в самих компактных моделях компьютеров типа subnotebook и palmtop конструкторы вынуждены идти на сокращения количества и размеров клавиш.

Расположение буквенных клавиш на компьютерных клавиатурах стандартно. Сегодня повсеместно применяется стандарт QWERTY - по первым шести латинским буквенным клавишам верхнего ряда. Ему соответствует отечественный стандарт QWERTY расположения клавиш кириллицы, практически аналогичный расположению клавиш на пишущей машинке.

Стандартизация в размере и расположении клавиш нужна для того, чтобы пользователь на любой клавиатуре мог без переучивания работать "слепым методом". Слепой десятипальцевый метод работы является наиболее продуктивным, профессиональным и эффективным. Увы, клавиатура из-за низкой производительности пользователя оказывается сегодня самым "узким местом" быстродействующей вычислительной системы.

Работать с клавиатурой очень просто и наглядно. Нажмите клавишу, и в компьютер перенесется код соответствующего символа. Нажатие одной или некоторой их определенной комбинации означает посылку в оперативную память одного или двух байтов информации. Чтобы каждому символу клавиатуры поставить в соответствие определенный байт информации, используют специальную таблицу кодов ASCII (American Standart Code for Information Interchange) - американский стандарт кодов для обмена информацией, применяемой на большинстве компьютеров. Таблица кодировки определяет взаимное соответствие изображений символов на экране дисплея с их числовыми кодами.

Заметим, что даже если название клавиш на клавиатуре совпадают, то их скэн-код все-таки различен, и поэтому в принципе это совершенно разные клавиши. Этот факт используется при написании специальных программ, определяющих реакцию процессора на нажатие определенной клавиши на клавиатуре.

После нажатия клавиши клавиатура посылает процессору сигнал прерывания и заставляет процессор приостановить свою работу и переключиться на программу обработки прерывания клавиатуры.

При этом клавиатура в своей собственной специальной памяти запоминает, какая клавиша была нажата (обычно в памяти клавиатуры может храниться до 20 кодов нажатых клавиш, если процессор не успевает ответить на прерывание). После передачи кода нажатой клавиши процессору эта информация из памяти клавиатуры исчезает.

Кроме нажатия клавиатура отмечает также и отпускание каждой клавиши, посылая процессору свой сигнал прерывания с соответствующим кодом. Таким образом компьютер "знает", держат клавишу или она уже отпущена. Это свойство используется при переходе на другой регистр. Кроме того, если клавиша нажата дольше определенного времени, обычно около половины секунды, то клавиатура генерирует повторные коды нажатия этой клавиши.

Ввод символов с клавиатуры осуществляется только в той точке экрана, где располагается курсор. Курсор представляет собой прямоугольник или черту контрастного цвета длинной в один символ.

Специальные клавиши клавиатуры

Специальные (служебные) клавиши выполняют следующие основные функции:

{ENTER} - ввод команд на выполнение процессором;

{ESC} - отмена какого-либо действия;

{TAB} - перемещение курсора на позицию табуляции;

{INS} - переключение режима вставки символа в положении курсора в режим забоя символа в положении курсора;

{DEL} - удаление символа в положении курсора;

{BACKSPACE} - удаление символа слева от курсора;

{HOME} - перемещение курсора в начало текста;

{END} - перемещение курсора в конец текста;

{PGUP} - перемещение курсора на одну экранную страницу по тексту вверх;

{PGDN} - перемещение курсора на одну экранную страницу по тексту вниз;

{ALT} и {CTRL} - при одновременном нажатии этих клавиш с какой-либо другой вызывается изменение действия последней;

{SHIFT} - удержание этой клавиши в нажатом состоянии обеспечивает смену регистра;

{CAPS LOCK} - фиксация/расфиксация регистра заглавных букв;

2. Мышка и другие манипуляторы

Хотя клавиатура еще вовсе не утратила значения для общения пользователя с компьютером, другое устройство ручного ввода информации - мышка - становится все более весомой и важной. Но даже рискуя сделать из мышки слона, можно уверено утверждать, что на современном компьютере работать без мышки почти невозможно: вы тут же увязните в графическом интерфейсе Windows и многих прикладных программах, работающих с окнами, меню, иконками и диалоговыми боксами.

Управлять курсором или маркером на экране с помощью одной клавиатуры бывает чудовищно неудобно, медленно и просто нелепо, когда для этого есть специальные устройства-указатели. Мышка и трекбол, которые "по-умному" принято называть координатными манипуляторами,- это самые распространенные сегодня устройства для дистанционного управления графическими изображениями на экране. В принципе, мышка и трекбол похожи на джойстик, известный всякому, кто увлекается компьютерными играми. Набирать какие-либо команды не нужно, достаточно при работе в программе указать мышкой нужную операцию меню или иконку в окне на экране, а затем щелкнуть кнопкой. Вот и все, что требуется, а остальное сделает программа.

Мышки бывают с двумя и тремя кнопками. Вообще-то практически для всех случаев жизни на мышке достаточно двух кнопок. Делом вкуса является также цвет и дизайн корпуса мышки. Выбор здесь огромный. Над этим старательно работают дизайнеры множества фирм, так что выбрать тут есть из чего.

Трекбол мало чем отличается от мышки. В сущности - это та же самая мышка, но перевернутая "вверх ногами", точнее - перевернутая вверх шаром. Если мышку надо возить по столу и, катая шарик, управлять перемещением маркера на экране, то в трекболе надо просто крутить пальцами или ладонью сам шарик в разные стороны.

В портативных компьютерах трекбол нередко встраивается прямо рядом с клавиатурой либо пристегивается с боку или спереди клавиатуры компьютера. Впрочем, и для настольных компьютеров выпускаются клавиатуры с "встроенным трекболом". А в самых портативных компьютерах вместо мышки и трекбола теперь используют крошечный пойнтер - небольшой цветной штырек, торчащий среди клавиш на клавиатуре, который, словно джойстик, можно нажимать в разные стороны.

А самый последний поиск мышиной моды в портативных компьютерах - в место пойнтера используется клавиша с буквой J. Это клавиша - или J-пойнтер - как раз и

служит таким джойстиком, воспринимающим нажатия в разные стороны, а окружающие клавишу J другие буквенные клавиши выполняют роль кнопок отсутствующей мышки или трекбола.

Мышки, вообще, как правило, более удобны, чем трекболы, но трекболы требуют меньше свободного места на рабочем столе. И если стол завален документами, книгами, чертежами, найти свободное место для мышки порой оказывается непросто. Кстати, шарик мышки катать не по голой поверхности стола, а по специальному резиново-пластиковому коврику. Тогда мышка меньше изнашивается и загрязняется, и указывает значительно точнее, а значит - быстрее работает и меньше утомляет глаза и руки пользователя.

Помимо традиционных мышек, подключенных к компьютеру тоненьким кабелем через последовательный порт или через специальный контроллер на плате расширения, некоторыми фирмами выпускаются перспективные беспроводные мышки. Ряд фирм выпускает мышки, передающих информацию с помощью инфракрасных лучей. Есть даже миниатюрные беспроводные мышки, которые надеваются на палец, словно перстень. А швейцарская фирма Logitech, признанный мировой лидер в этой области, выпустила мышку, связанную с компьютером по радио. Впрочем, это довольно дорогие устройства, нужны далеко не каждому пользователю.

Самым изысканным эстетическим и техническим требованиям отвечают сегодня мышки и трекболы фирм Microsoft и Logitech. Фактическим стандартом в мышиной технологии является мышка Microsoft Mouse. Мышки и трекболы всех остальных фирм ориентируются на этот стандарт.

3. Принципы ввода информации с бумажных носителей

Ввод графической информации в ЭВМ для АСУ производится в три этапа. На первом этапе определяются координаты графических элементов, на втором - координаты преобразуются в цифровой код, на третьем - они записываются в память ЭВМ и передаются для обработки в арифметическое устройство (АУ).

Определение координат графических элементов можно производить автоматическим и полуавтоматическим способами. Преобразование координат графических элементов в цифровой код осуществляется несколькими методами:

1. в память ЭВМ записываются значения текущих координат всех элементов;

2. графическая информация представляется в аналитическом виде;

3. исходные данные описываются на специальном графическом языке.

Все перечисленные методы и способы преобразования и представления в ЭВМ графической информации определяют требования, предъявляемые к техническим средствам преобразования информации для ЭВМ в АСУ.

Устройство ввода графической информации (УВГИ) - это устройство, преобразующее графические данные в машинные коды.

Любую графическую информацию можно рассматривать как набор оптических неоднородностей, отличающихся по яркости и цвету. Таким образом, любое УВГИ решает следующие задачи:

1. дискретизация изображения на элементы;

2. преобразование оптической информации в электрический аналоговый сигнал;

3. преобразование аналогового сигнала в цифровой код.

Количество дискретных элементов определяется заданной точностью представления графической информации. Объемом информации о графическом изображении определяется быстродействие УВГИ.

По методам дискретизации различают УВГИ автоматического и полуавтоматического типов. К автоматическим УВГИ относятся матричные, сканирующие и следящие устройства; к полуавтоматическим - телевизионные, акустические, оптические, электрические и электромеханические устройства.

4. Сканер

Вводить изображение в компьютер можно разными способами, например, используя видеокамеру или цифровую фотокамеру. Еще одним устройством ввода графической информации в компьютер является оптическое сканирующее устройство, которое обычно называют сканером. Сканер позволяет оптическим путем вводить черно-белую или цветную печатную графическую информацию с листа бумаги. Отсканировав рисунок и сохранив его в виде файла на диске, можно затем вставить его изображение в любое место в документе с помощью программы текстового процессора или специальной издательской программы электронной верстки, можно обработать это изображение в программе графического редактора или отослать изображение через факс-модем на телефакс, находящейся на другом конце света.

Сканер - это глаза компьютера. Первоначально они создавались именно для ввода графических образов, рисунков, фотоснимков, чертежей, схем, графиков, диаграмм. Однако, помимо ввода графики, в настоящее время они все шире используются в довольно сложных интеллектуальных системах OCD или Optical Character Recognition, то есть оптического распознания символов. Эти "умные" системы позволяют вводить в компьютер и читать текст.

Сперва текст вводится в компьютер с бумаги как графическое изображение. Затем компьютерная программа обрабатывает это изображение по сложным алгоритмам и превращает в обычный текстовый файл, состоящий из символов ASCII. А это значит, что текст книги или газетной статьи можно быстро вводить в компьютер, вовсе не пользуясь клавиатурой!

А если система распознавания OCR соединяется еще и с программой перевода, в компьютер можно вводить страницы текста на иностранном языке и почти мгновенно получать готовый перевод. Конечно, литературные качества электронного перевода обычно не слишком высокие, в научно-технических текстах литературные достоинства - не самое главное, зато готовый перевод формально достаточно точен, и его можно получить фантастически быстро.

Сканеры бывают различных конструкций.

Ручной сканер - это самый простой и дешевый сканер. Ручной сканер, словно мышка, соединяется кабелем с компьютером. При прокатывании сканера по странице книги или журнала, необходимое изображение считывается и в цифровом коде вводиться в память компьютера. В ручном сканере роль привода считывающего механизма выполняет рука. Понятно, что равномерность перемещения сканера существенно сказывается на качестве вводимого в компьютер изображения. Ширина вводимого изображения для ручных сканеров обычно не превышает 4 дюймов (10 см). Современные ручные сканеры могут обеспечивать автоматическую "склейку" изображения, то есть формируют целое изображение из отдельно вводимых его частей. К основным достоинствам этих сканеров относятся небольшие габаритные размеры и сравнительно низкая цена, однако добиться высокого качества изображения с их помощью очень трубно, поэтому ручные сканеры можно использовать для

ограниченного круга задач. Кроме того они совершенно лишены "интеллектуальности", свойственной другим типам сканеров.

Планшетный сканер. Это наиболее распространенный тип сканеров. Первоначально он использовался для сканирования непрозрачных оригиналов. Почти все модули имеют съемную крышку, что позволяет сканировать "толстые" оригиналы (журналы, книги). Дополнительно некоторые модели могут оснащаться механизмом подачи отдельных листов, что удобно при работе с программами распознавания текстов - OCR (Optical Characters Recognition). В последние время многие фирмы-лидеры в производстве плоскостных сканеров стали дополнительно предлагать слайд-модуль (для сканирования прозрачных оригиналов). Слайд-модуль имеет свой, расположенный сверху, источник света. Такой слайд-модуль устанавливается на плоскостной сканер вместо простой крышки и превращает сканер в универсальный (плоскостной сканер с установленным слайд-модулем).

Барабанный сканер. Основное его отличие состоит в том, что оригинал закрепляется на прозрачном барабане, который вращается с большой скоростью. Считывающий элемент располагается максимально близко от оригинала. Данная конструкция обеспечивает наибольшее качество сканирования. Обычно в барабанные сканеры устанавливают три фотоумножителя, и сканирование осуществляется за один проход. "младшие" модели у некоторых фирм с целью удешевления используют вместо фотоумножителя фотодиод в качестве считывающего элемента. Барабанные сканеры способны сканировать любые типы оригиналов.

В отличие от плоскостных сканеров со слайд-модулем, барабанные могут сканировать непрозрачные и прозрачные оригиналы одновременно.

Проекционный сканер. Этот тип сканеров применяется для сканирования с высоким разрешением и качеством слайдов небольшого формата (как правило, размером не более 4?5 дюймов). Существует две модификации: с горизонтальным и вертикальным расположением оптической оси считывания. Наиболее популярным в России, как, впрочем, и на Западе, является вертикальный проекционный сканер.

Типов оригиналов бывает всего два. Это прозрачные негативные и позитивные слайды, которые сканируют в проходящем свете. Непрозрачные оригиналы представляют собой либо аналоговые изображения - фотографии, либо дискретные - иллюстрации из печатных изданий (в полиграфии полутоновая печать осуществляется с помощью растровых точек различного цвета и размера).

Считывание изображения. Механизмы считывания изображения базируются или на фотоумножителе, или на ПЗС. Фотоумножитель проще всего сравнить с радиолампой-фотосенсором, у которой имеются пластины катода и анода и которая конвертирует свет в электрический сигнал. Считываемая информация подается на фотоумножитель точка за точкой с помощью засвечивающего луча. ПЗС - относительно дешевый полу проводниковый элемент довольно малого размера. ПЗС так же как и умножитель конвертирует световую энергию в электрический сигнал. Набор элементарных ПЗС-элементов располагают последовательно в линию, получая линейку для считывания сразу целой строки, естественно и освещается сразу целая строка оригинала. Цветное изображение такими сканерами считывается за три прохода (с помощью RGB-светофильтра). Многие сканеры имеют три параллельные линейки ПЗС, тогда сканирование цветных оригиналов осуществляется за один проход, так как каждая линейка считывает один из трех базовых цветов. Потенциально ПЗС-сканеры более быстродейственны, чем барабанные сканеры на фотоумножителях.

Качество изображения. Сканеры различаются по многим параметрам - по технологии считывания изображения, типу механизма и некоторым другим. Существуют

параметры сканирующего устройства, влияющие на качество изображения. К таким параметрам относится оптическая разрешающая способность, число передаваемых полутонов и цветов, диапазон оптических плотностей, интеллектуальность сканера, световые искажения, точность фокусировки (резкость).

Интеллектуальность сканера. Под интеллектуальностью обычно подразумевается способность сканера с помощью заложенных в нем аппаратным и поставляемых с ним программных средств автоматически настраиваться и минимизировать потери качества. Наиболее ценятся сканеры, обладающие способностью автокалибровки, т.е. настройки на динамический диапазон плотностей оригинала, а также компенсации цветовых искажений. Допустим, мы имеем ПЗС-сканер, воспринимающий оптический диапазон плотностей до 3.2. С его помощью нам нужно отсканировать слайд, имеющий максимальную оптическую плотность 4.0. "Хороший" сканер сначала делает предварительное сканирование для анализа оригинала и получения диаграммы оптических плоскостей. После анализа диаграммы сканер производит свою автокалибровку с целью сдвига своего динамического диапазона восприятия оптических плотностей. Таким образом минимизируются потери в "тенях" благодаря сокращению потерь в "светах".

Цветовые искажения сканеров. Каждый сканер обладает своими собственными недостатками при восприятии цветов и общими недостатками, присущими данной модели. Общие недостатки обусловлены техническими возможностями и механическими характеристиками модели. Собственный недостаток сканера обусловлен индивидуальной способностью освещающего оригинал источника света и считывающего элемента. Считается, что все продаваемые сканеры проходят заводскую калибровку. Однако, если сканер имеет функцию автокалибровки, то это большое преимущество перед сканером, лишенным такой функции. Автокалибровка сканера позволяет скорректировать цветовые искажения и увеличить число распознаваемых цветовых оттенков. Поскольку источник света имеет свойство изменять свои характеристики со временем, как, впрочем, и считывающий элемент, наличие автокалибровки приобретает первостепенное значение, если Вы постоянно с цветными полутоновыми изображениями. Практически все современные модели сканеров обладают такой функцией.

5. Цифровая фотокамера

Чтобы ввести цветное изображение со снимка в память компьютера, нужен цветной сканер или дигитайзер для ввода слайдов.

Спрашивается, а нужно ли вообще вводить изображения в компьютер? Убедительных аргументов в пользу ввода снимков в компьютер может быть немало. Во-первых, для профессиональных целей фоторепортерам порой действительно нужны мгновенные снимки, чтобы сразу же убедиться в их качестве и выразительности. Во-вторых, такие цифровые снимки можно немедленно использовать для электронной верстки, например, в журналисткой практике в газете, на телевидении или в информационном агентстве. В-третьих, файл с изображением можно тут же переправить по каналам связи на любое расстояние. В-четвертых, в цифровые изображения в компьютере можно легко вмешиваться, их удобно редактировать, кадрировать, ретушировать, оснащать спецэффектами. В-пятых, вполне оправдан повсеместный отказ от применения химических процессов по экологическим соображениям. В-шестых, долговременно хранить готовые фотоснимки удобнее и надежнее на компакт-дисках. В-седьмых, с помощью компьютера весьма удобно показывать снимки в большой аудитории, студентам или школьникам. В-восьмых, цифровые снимки необходимы для создания мультимедиа. И еще многое другое.

Итак, цифровая камера предназначена для ввода изображений в компьютер. Но печатные изображения в компьютер можно ввести и с помощью сканера, а "живые" кадры можно "схватить" и ввести прямо с видеокамеры или с видеомагнитофона. Однако цифровые фотокамеры превосходят по качеству ввод с видеокамеры. Кроме того, цифровая камера - самый быстрый и простой способ ввода изображения в компьютер. Цифровые камеры записывают изображение в память, которая затем может быть без дополнительных специальных устройств введена в любой компьютер через порт связи.

А чтоб навсегда сохранить полученные снимки, фирма Kodak разработала практическую и недорогую технологию размещения электронных фотографий на компакт-дисках в стандарте Kodak Photo CD. Эта технология скоро вытеснит традиционную химическую фотографию. На каждом компакт-диске может поместиться целый фотоальбом. С помощью плеера, диски Photo CD можно просматривать на экране любого телевизора или компьютера.

6. Дигитайзер

Дигитайзер - это еще одно устройство ввода графической информации, имеющее пока сравнительно узкое применение для некоторых специальных целей. Свое название дигитайзеры получили от английского digit - «цифра». То есть по-русски их можно назвать просто "оцифровыватели". Впрочем, есть и более благозвучное название - англо-цифровые преобразователи.

Обычно дигитайзеры выполняются в виде планшета. Поэтому такие устройства часто называют графическими планшетами. Применяется такой дигитайзер для поточечного координатного ввода графических изображений в системах автоматического проектирования, в компьютерной графике и анимации. Надо отметить, что это далеко не самый быстрый и удобный способ построения рисунков и чертежей, особенно в случае сложной геометрии. Но зато графический планшет обеспечивает наиболее точный ввод графической информации в компьютер.

Графический планшет обыкновенно содержит рабочую плоскость, рядом с которой находятся кнопки управления. На рабочую плоскость может быть нанесена вспомогательная координатная сетка, облегчающая ввод сложных изображений в компьютер. Для ввода информации служит специальное перо или координатное устройство с "прицелом", подключенное кабелем к планшету. Сам дигитайзер также подключается к компьютеру кабелем через порт связи. Разрешающая способность таких графических планшетов не менее 100 dpi (точек на дюйм).

В самых совершенных и дорогих дигитайзерах ввод информации происходит без специальных перьев или прицелов, так как рабочая поверхность планшета обладает "тактильной чувствительностью", основанной на использовании пьезоэлектрического эффекта. При нажатии на точку, расположенную в приделах рабочей поверхности планшета, под которой проложена сетка из тончайших проводников, на пластине пьезоэлектрика возникает разность потенциалов. Координаты этой точки обнаруживаются программой-драйвером, сканирующей сетку проводников. Эта программа выполнит отображение точки на экран монитора. Пьезоэлектрические дигитайзеры позволяют чертить на рабочей поверхности планшета, словно на обычной чертежной доске, и таким образом вводить даже несуществующие изображения. При этом графическая информация вводится с разрешением 400 dpi.

Кстати говоря, на этом же принципе основаны новые координатные устройства для работы в графическом интерфейсе пользователя (в операционной среде Windows или OS/2), предназначенные для замены традиционных мышек и трекболов. Всякий, кто пробовал воспользоваться такими тактильными устройствами, изготовленными,

например, японской фирмой Toshida, мог убедиться, что гораздо удобнее и легче водить пальцем по окошку дигитайзера размером менее спичечной коробки, чем пользоваться обычной мышкой: курсор на экране весьма послушно и чутко повторяет движения пальца на планшете. Ни каких дополнительных кнопок в таком дигитайзере нет. Указав на экране дисплея нужный выбор, достаточно дважды стукнуть пальцем по окошку и компьютер поймет сообщение.

Для ввода графической информации могут так же использоваться некоторые виды планшетных графопостроителей. Однако многие готовые изображения (фотографии, чертежи, рисунки, карты, графики, слайды, кино-фильмы) гораздо удобнее вводить с помощью специального видеодигитайзера. В простейшем случае видеодигитайзером может даже служить видео-камера. В настоящее время выпускается множество специальных графических систем с различными типами видеодигитайзеров, позволяющих вводить в компьютер цветные изображения с бумаги или со слайдов. К числу видеодигитайзеров относится и цифровая фотокамера.

В современных киностудиях применяются специальные дигитайзеры для переноса изображения с кинопленки в компьютер. После цифровой обработки изображение снова помещается на пленку. В связи с этим поговаривают, что скоро компьютеры смогут вообще вытеснить из кино живых актеров. Такое предположение вполне реально. Например, в компьютер введут фотографии кинозвезд, компьютер синтезирует из этих снимков некий произвольный персонаж, который своим обликом будет точно соответствовать вкусам зрителей. Затем этот синтетический герой может очень правдоподобно "ожить" на экране, и при этом совершать невероятные трюки, словно персонаж мультипликации.

Дигитайзером в компьютерах киностудий уже сегодня вводят фотографии пейзажей и нарисованные декорации, интерьеры и костюмы. Надвигается эпоха виртуальной реальности, созданной в памяти компьютера.

3.13 Практическое занятие № 13 (4 часа)
Тема: «Виды архитектур ЛВС»
3.13.1 Задачи работы:
1. Научиться разбираться в ЛВС
3.13.2 Краткое описание практического:
ЛВС с выделенным сервером
Одноранговые ЛВС
3.13.3Результаты и выводы:

1. ЛВС с выделенным сервером.

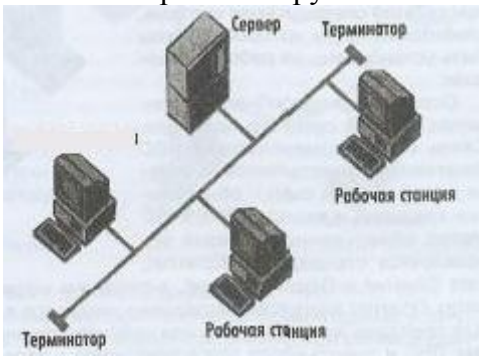
При выборе компьютера на роль файлового сервера необходимо учитывать следующие факторы:
• быстродействие процессора;
• скорость доступа к файлам, размещенным на жестком диске;
• емкость жесткого диска;
• объем оперативной памяти;
• уровень надежности сервера;
• степень защищенности данных.

Возникает вопрос, зачем файл-серверу высокое быстродействие, если прикладные программы выполняются на рабочих станциях? Во время работы большой ЛВС файловый сервер обрабатывает огромное количество запросов на обслуживание файлов, а на это

затрачивается значительное процессорное время. Для того, чтобы ускорить обслуживание запросов и создать у пользователя впечатление, что именно он является единственным клиентом сети, необходим быстродействующий процессор.
Но все же наиболее важным компонентом файлового сервера является дисковый накопитель. На нем хранятся все файлы пользователей сети. Быстрота доступа, емкость и надежность накопителя во многом определяют, насколько эффективным будет использование сети.
Сетевые ОС с выделенным файл-сервером обычно имеют более высокую производительность, поскольку они оптимизированы именно под выполнение операций с файлами. В принципе, никаких более важных действий на выделенном файл-сервере не выполняется. Значительного повышения производительности работы сервера можно добиться, увеличивая его оперативную память. В одноранговой сети 128 мегабайт памяти может быть вполне достаточно, в то время как для крупной сети с выделенным файл-сервером желательна память объемом 512 и более мегабайт. Если файловый сервер снабжен оперативной памятью достаточного объема, то он имеет возможность именно в оперативной памяти хранить те области дискового пространства, *к* которым обращаются наиболее часто. Такой метод хорошо известен, часто применяется для ускорения доступа к данным на обычных ПК и называется методом кэширования. Ведь если идет обращение к файлу, данные которого в данный момент находятся в кэше, сервер может передать искомую информацию, не обращаясь к диску. В результате этого будет достигнут значительный временной выигрыш.
Сетевой адаптер, установленный на файловом сервере - это такое устройство, через которое проходят практически все данные, функционирующие в локальной сети. В связи с этим необходимо, чтобы этот адаптер работал быстро. Сетевой адаптер становится более быстродействующим в результате, во-первых, повышения его разрядности и, во-вторых, увеличения объема его собственного ОЗУ. На файл-сервере должен быть установлен сетевой адаптер для шины PCI, что позволяет поддерживать высокую скорость передачи данных.

2. Одноранговые ЛВС.

В одноранговых сетях любой компьютер может быть и файловым сервером, и рабочей станцией одновременно. Преимущество одноранговых сетей заключается в том, что нет необходимости копировать все используемые сразу несколькими пользователями файлы на сервер. В принципе любой пользователь сети имеет возможность использовать все данные, хранящиеся на других компьютерах сети, и устройства, подключенные к ним. Основной недостаток работы одноранговой сети заключается в значительном увеличении времени решения прикладных задач. Это связано с тем, что каждый компьютер сети отрабатывает все запросы, идущие к нему со стороны других пользователей. Следовательно, в одноранговых сетях каждый компьютер работает значительно интенсивнее, чем в автономном режиме.
Затраты на организацию одноранговых вычислительны сетей относительно небольшие, Однако при увеличении числа рабочий станций эффективность их использования резко уменьшается Пороговое значение числа рабочих станций составляет, по оценкам фирмы Novell, 25-30. Поэтому одноранговые сети используются только для относительно небольших рабочих групп.

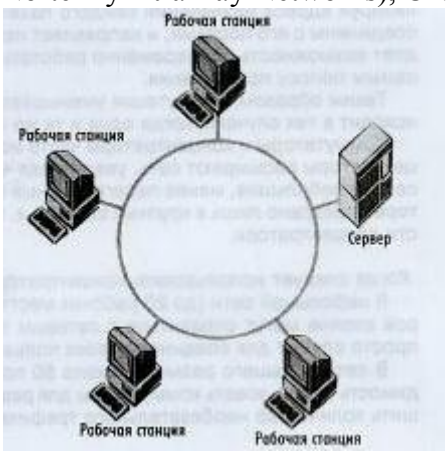


Архитектура ЛВС.
Различают три наиболее распространенные сетевые архитектуры, которые используются и для одноранговых сетей и для сетей с выделенным файл-

сервером. Это так называемые шинная, кольцевая и звездообразная структуры.
В случае реализации шинной структуры все компьютеры связываются в цепочку. Причем на ее концах надо разместить так называемые терминаторы, служащие для гашения сигнала. Если же хотя бы один из компьютеров сети с шинной структурой оказывается неисправным, вся сеть в целом становится неработоспособной. В сетях с шинной архитектурой для объединения компьютеров используется тонкий и толстый кабель. Максимальная теоретически возможная пропускная способность таких сетей составляет 10 Мбит/с, Такой пропускной способности для современных приложений, использующих видео- и мультимедийные данные, явно недостаточно, Поэтому почти повсеместно применяются сети с звездообразной архитектурой.
Для построения сети с звездообразной архитектурой в центре сети необходимо разместить концентратор. Его основная функция - обеспечение связи между компьютерами, входящими в сеть. То есть все компьютеры, включая файл-сервер, не связываются непосредственно друг с другом, а присоединяются к концентратору. Такая структура надежнее, поскольку в случае выхода из строя одной из рабочих станций все остальные сохраняют работоспособность. В сетях же с шинной топологией в случае повреждения кабеля хотя бы в одном месте происходит разрыв единственного физического канала, необходимого для движения сигнала. Кроме того, сети с звездообразной топологией поддерживают технологии Fast Ethernet и Gigabit Ethernet, что позволяет увеличить пропускную способность сети в десятки и даже сотни раз (разумеется при использовании соответствующих сетевых адаптеров и кабелей).
Кольцевая структура используется в основном в сетях Token Ring и мало чем отличается от шинной. Также в случае неисправности одного из сегментов сети вся сеть выходит из строя. Правда, отпадает необходимость в использовании терминаторов.

В сети любой структуры в каждый момент времени обмен данными может происходить только между двумя компьютерами одного сегмента. В случае ЛВС с выделенным файл-сервером - это файл-сервер и произвольная рабочая станция; в случае одноранговой ЛВС - это любые две рабочие станции, одна из которых выполняет функции файл-сервера. Упрощенно диалог между файл-сервером и рабочей станцией выглядит так: открыть файл - подтвердить открытие файла; передать данные файла - пересылка данных; закрыть файл – подтверждение закрытия файла. Управляет диалогом сетевая операционная система, клиентские части которой должны быть установлены на рабочих станциях.
Остановимся подробнее на принципах работы сетевого адаптера. Связь между компьютерами ЛВС физически осуществляется на основе одной из двух схем - обнаружения коллизий и передачи маркера. Метод обнаружения коллизий используется стандартами Ethernet, Fast Ethernet и Gigabit Ethernet, а передачи маркера - стандартом Token Ring. В сетях Ethernet адаптеры непрерывно находятся в состоянии прослушивания сети. Для передачи данных сервер или рабочая станция должны дождаться освобождения ЛВС и только после этого приступить к передаче. Однако не исключено, что передача может начаться несколькими узлами одного сегмента сети одновременно, что приведет к коллизии. В случае возникновения коллизии, узлы должны повторить свои сообщения. Повторная передача производится адаптером самостоятельно без вмешательства процессора компьютера. Время, затрачиваемое на преодоление коллизии, обычно не превышает одной микросекунды. Передача сообщений в сетях Ethernet производится пакетами со скоростью 10, 100 и 1000 Мбит/с. Естественно, реальная загрузка сети меньше, поскольку требуется время на подготовку пакетов. Все узлы сегмента сети

принимают сообщение, передаваемое компьютером этого сегмента, но только тот узел, которому оно адресовано, посылает подтверждение о приеме. Основными поставщиками оборудования для сетей Ethernet являются фирмы 3Com, Bay Networks (недавно компания Nortel купила Bay Networks), CNet.



В ЛВС с передачей маркера сообщения передаются последовательно от одного узла к другому вне зависимости от того, какую архитектуру имеет сеть - кольцевую или звездообразную. Каждый узел сети получает пакет от соседнего. Если данный узел не является адресатом, то он передает тот же самый пакет следующему узлу. Передаваемый пакет может содержать либо данные, направляемые от одного узла другому, либо маркер. Маркер - это короткое сообщение, являющееся признаком незанятости сети. В том случае, когда рабочей станции необходимо передать сообщение, ее сетевой адаптер дожидается поступления маркера, а затем формирует пакет, содержащий данные, и передает этот пакет в сеть. Пакет распространяется по ЛВС от одного сетевого адаптера к другому до тех пор, пока не дойдет до компьютера-адресата, который произведет в нем стандартные изменения. Эти изменения являются подтверждением того, что данные достигли адресата. После этого пакет продолжает движение дальше по ЛВС, пока не возвратится в тот узел, который его сформировал. Узел - источник убеждается в правильности передачи пакета и возвращает в сеть маркер. Важно отметить, что в ЛВС с передачей маркера функционирование сети организовано так, что коллизии возникнуть не могут. Пропускная способность сетей Token Ring равна 16 Мбит/с. Оборудование для сетей Token Ring производит IBM, 3Com и некоторые другие фирмы.

3.14 Практическое занятие № 14 (4 часа)
Тема: «Архитектура Ethernet»
3.14.1 Задачи работы:
1. Научиться разбираться в архитектуре Ethernet
3.14.2 Краткое описание практического:
Архитектура Ethernet
Гигабитный Ethernet
10-Гигабитный Ethernet
3.14.3Результаты и выводы:

1. Архитектура Ethernet

Популярность Ethernet нередко вызывает удивление. Эта технология изначально не является эффективной. На самом деле только 37 % полосы пропускания подходит для ее функционирования, так как Ethernet работает в условиях одновременного использования канала связи. Устройства, подключенные к локальной сети Ethernet, прослушивают линию и ожидают ее освобождения для отправки сообщения. Если два устройства одновременно начинают передачу данных, и их пакеты сталкиваются, то обе передачи прерываются, и рабочие станции через некоторое время, определяемое случайным образом, осуществляют новую попытку отправки данных.

Ethernet использует алгоритм CSMA/CD (Carrier Sense Multiple Access with Collision Detection - множественный доступ с контролем несущей и обнаружением конфликтов) для прослушивания линии, распознавания коллизии и прерывания передачи. CSMA/CD является "светофором" технологии Ethernet и служит для предотвращения беспорядочных столкновений пакетов в сети. На рисунке 1.5 показано, как работает алгоритм CSMA/CD.

Рис. 1.5. Работа алгоритма CSMA/CD
Технология Ethernet использует общую среду передачи, поэтому все устройства локальной сети Ethernet получают все сообщения, а затем проверяют, совпадает ли адрес назначения с собственным адресом устройства. Если адреса совпадают, то сообщение принимается и проходит через все семь уровней стека, в противном случае сообщение отбрасывается.

Реализация коммутируемой архитектуры сети Ethernet имеет преимущество в том, что линии, связывающие коммутатор с устройствами, подключенными к сети, получают полосу пропускания максимальной ширины. Это объясняется тем, что передаваемые пакеты не отправляются широковещанием ко всем устройствам сети, а передаются от коммутатора к пункту назначения.

2. Гигабитный Ethernet

Гигабитный Ethernet является расширением Ethernet-стандарта до скорости 1000 Мб/с. Такой скачок вызван тем, что он наследует возможности других Ethernet-спецификаций (исходного 10-мегабитного Ethernet и 100-мегабитного Fast Ethernet).

Технология гигабитного Ethernet является основным конкурентом технологии ATM. (Об этой технологии мы будем кратко говорить далее). Так как Ethernet является самой популярной сетевой технологией, то гигабитный Ethernet выигрывает у ATM, поскольку является более изученным. Изначально он разрабатывался для эксплуатации в локальных сетях, но при возрастании скорости передачи данных до 1 Гбит/с его можно использовать в качестве WAN-технологии.

Несмотря на высокую скорость передачи, Ethernet не совсем подходит для глобальных сетей. Эта технология использует кадры переменного размера - от 64 до 1400 байт, что не соответствует характеристикам ATM по качеству обслуживания (Quality of Service, QoS).

Примечание. Качество обслуживания (QoS) гарантирует наиболее эффективную отправку и получение пакетов. В "Подключения к рабочим группам" мы более подробно рассмотрим QoS и его реализацию в Windows XP.

Разумеется, о конкретных нуждах и эксплуатационных возможностях организации можно говорить долго. Если компания не делает акцент на характеристики QoS и имеет достаточную базу знаний об Ethernet, то идеальным решением для нее является гигабитный Ethernet. Популярным решением такой сети является подключение локальных сетей общего доступа с технологией Fast Ethernet к магистральной сети, работающей по технологии гигабитного Ethernet.

3. 10-гигабитный Ethernet

Следующей ступенькой в развитии Ethernet стал 10-гигабитный Ethernet. Скорости передачи 10 Мбит/с и 100 Мбит/с в технологии Ethernet делают ее хорошим способом доступа к данным, гигабитный Ethernet становится претендентом на роль WAN. А 10-гигабитная реализация Ethernet становится настоящей глобальной сетевой технологией.

10-гигабитный Ethernet использует Ethernet-протокол, формат кадра и размер кадра, определенные в спецификации IEEE 802.3. Переменный размер кадров по-прежнему остается проблемой, однако принципиальных трудностей для группировки множества мелких пакетов в один большой транк (trunk) технологии 10-гигабитного Ethernet, не существует. Все "за" и "против" должны рассматриваться для конкретной организации или ситуации.

3.15 Практическое занятие № 15 (4 часа)
Тема: «Многомашинные и многопроцессорные вычислительные системы»
3.15.1 Задачи работы:
1. Освоить многомашинные и многопроцессорные вычислительные системы
3.15.2 Краткое описание практического:
Многомашинные вычислительные системы
Многопроцессорные вычислительные системы
3.15.3Результаты и выводы:

1. Многомашинные вычислительные системы

Многомашинная вычислительная система (ММВС) – система (комплекс), включающая в себя две или более ЭВМ (каждая из которых имеет процессор, ОЗУ, набор периферийных устройств и работает под управлением собственной ОС), связи между которыми обеспечивают выполнение функций, возложенных на ММВС.

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

В *косвенно-,* или *слабосвязанных* ММВС ЭВМ связаны друг с другом только через внешние запоминающие устройства (ВЗУ). Структурная схема такого ММВС приведена на рис. 3.1. (при трех и более ЭВМ комплексы строятся аналогичным образом). В косвенно-связанных системах связь между ЭВМ осуществляется только на информационном уровне. Такая организация связей обычно используется в тех случаях,

когда необходимо повысить надежность комплекса путем резервирования ЭВМ. В этом случае может быть несколько способов организации работы ММВС:

- Резервная ЭВМ находится в выключенном состоянии (ненагруженный резерв) и включается только при отказе основной ЭВМ.
- Резервная ЭВМ находится в состоянии полной готовности и в любой момент может заменить основную ЭВМ (нагруженный резерв), причем либо не решает никаких задач, либо работает в режиме самоконтроля, решая контрольные задачи.
- Для того чтобы полностью исключить перерыв в выдаче результатов, обе ЭВМ, и основная и резервная, решают одновременно одни и те же задачи, но результаты выдаст только основная ЭВМ, а в случае выхода се из строя результаты начинает вы давать резервная ЭВМ.

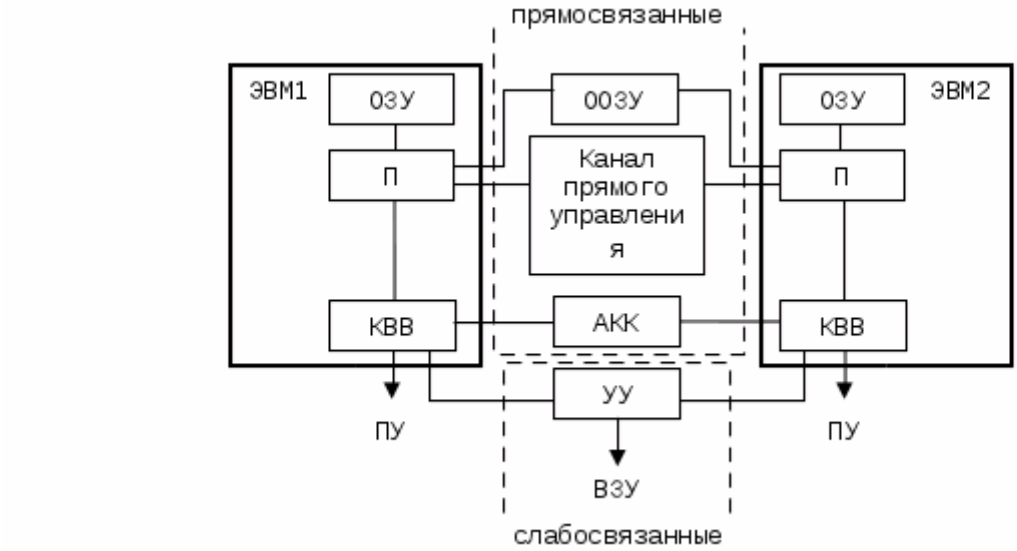


Рис. 3.1. Связи ЭВМ в составе ММВС

Прямосвязанные ММВС обладают существенно большой гибкостью. В ММВС существуют три вида связей (рис. 3.1): общее ОЗУ (ООЗУ); прямое управление, иначе связь процессор – процессор; адаптер канал – канал (АКК).

Связь через ООЗУ значительно сильнее связи через ВЗУ, вследствие того, что процессоры имеют прямой доступ к ОЗУ, хотя тоже информационная.

Непосредственная связь между процессорами – канал прямого управления – может быть не только информационной, но и командной, что, естественно, улучшает динамику перехода от основной ЭВМ к резервной и позволяет осуществлять более полный взаимный контроль ЭВМ.

Связь через адаптер канал – канал обеспечивает достаточно быстрый обмен информацией между ЭВМ, при этом обмен может производиться большими массивами информации. В отношении скорости передачи информации связь через АКК мало уступает связи через общее ОЗУ, а в отношении объема передаваемой информации – связи через общее ВЗУ.

Прямосвязанные ММВС позволяют осуществлять все способы организации работы ММВС, характерные для слабосвязанных ММВС, но значительно более эффективно.

Для ММВС с *сателлитными связями ЭВМ* характерным является не способ связи, а принципы взаимодействии ЭВМ. Структура связей в сателлитных ММВС не отличается от вышерассмотренных (чаще используется АКК). Особенностью этих ММВС является то, что в них, во-первых, ЭВМ существенно различаются по своим характеристикам, а во-вторых, имеет место определенная соподчиненность машин и различие функций, выполняемых каждой ЭВМ. Основная ЭВМ (чаще более высокопроизводительная) предназначена для основной обработки информации. Сателлитная (подчиненная меньшей

производительности) осуществляет организацию обмена информацией основной ЭВМ с периферийными устройствами, ВЗУ, удаленными абонентами и т.д. Некоторые ММВС могут включать не одну, а несколько сателлитных ЭВМ, при этом каждая из них ориентируется на выполнение определенных функций.

Сателлитные ММВС значительно увеличивают производительность, не оказывая заметного влияния на показатели надежности.

2. Многопроцессорные вычислительные системы

Многопроцессорная вычислительная система (МПВС) – это система (комплекс), включающий в себя два или более процессоров, имеющих общую ОП, общие периферийные устройства и работающих под управлением единой ОС, которая, в свою очередь, осуществляет общее управление техническими и программными средствами комплекса. При этом каждый из процессоров может иметь индивидуальные, доступные только ему ОЗУ и периферийные устройства.

Следует отметить, что МПВС в аппаратном плане значительно более сложны чем ММВС. При этом основная функция по организации вычислительного процесса возлагается на ОС, что значительно осложняет ее построение.

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

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

Существует три типа структурной организации МПВС: с общей шиной; с перекрестной коммутацией; с многовходовыми ОЗУ.

В МПВС с *общей шиной* проблема связей всех устройств между собой решается крайне просто: все они соединяются общей шиной, по которым передаются информация, адреса и сигналы управления (рис. 3.2). Интерфейс является односвязным, т. е. обмен информацией в любой момент времени может происходить только между двумя устройствами. Если потребность в обмене существует более чем у двух устройств, то возникает конфликтная ситуация, которая разрешается с помощью системы приоритетов и организации очередей в соответствии с этим. Обычно функции арбитра выполняет либо процессор, либо специальное устройство, которое регистрирует все обращения к общей шине и распределяет шину во времени между всеми устройствами комплекса.

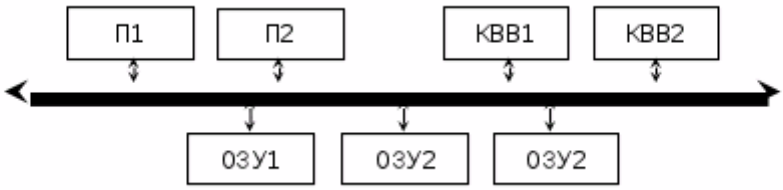


Рис. 3.2. МПВС с общей шиной

Достоинством такой структуры является простота, в том числе изменения комплекса, а также доступность модулей ОЗУ для всех остальных устройств.

Недостатками является невысокое быстродействие (одновременный обмен информацией возможен между двумя устройствами, не более), относительно низкая надежность системы из-за наличия общего элемента – шины.

МПВС с перекрестной коммутацией лишены недостатков, присущих МПВС с общей шиной. В таких МПВС все связи между устройствами осуществляются с помощью коммутационной матрицы (рис. 3.3.). Коммутационная матрица (КМ) позволяет связывать друг с другом любую пару устройств, причем таких пар может быть сколько угодно: связи не зависят друг от друга.

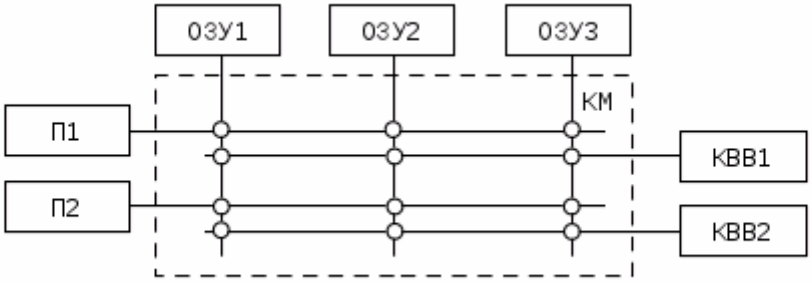


Рис. 3.3. МПВС с перекрестной коммутацией

В МПВС с перекрестной коммутацией возможность одновременной связи нескольких пар устройств позволяет добиваться очень высокой производительности комплекса.

Кроме того, к достоинствам структуры с перекрестной коммутацией можно отнести простоту и унифицированность интерфейсов всех устройств, а также возможность разрешения всех конфликтов в коммутационной матрице. Важно отметить и то, что нарушение какой-то связи приводит не к выходу из строя всего комплекса, а лишь к отключению какого-либо устройства, т. е. надежность таких комплексов достаточно высока.

Недостатками таких МПВС является сложность наращивания, что требует установки новой коммутационной матрицы, а также то, что при большой номенклатуре устройств КМ становится сложной, громоздкой и достаточно дорогостоящей.

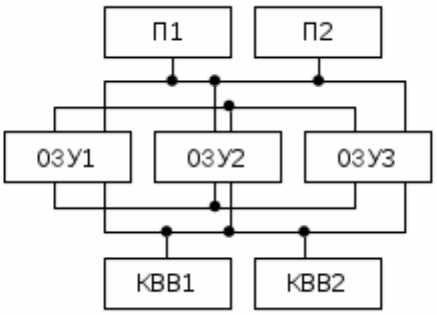


Рис. 3.4. МПВС с многовходовым ОЗУ

В *МПВС с многовходовыми ОЗУ* все, что связано с коммутацией устройств, осуществляется в ОЗУ. В этом случае модули ОЗУ имеют число входов, равное числу устройств, которые к ним подключаются. Структура такого МПВС показана на рис. 3.4.

В отличие от МПВС с перекрестной коммутацией, которые имеют централизованное коммутационное устройство, в МПВК с многовходовыми ОЗУ средства коммутации распределены между несколькими устройствами. Такой способ организации МПВК сохраняет все преимущества систем с перекрестной коммутацией, несколько упрощая при этом саму систему коммутации.

Кроме приведенных структурных организаций ММВС и МПВС нередко встречаются и смешанные.

3.16 Практическое занятие № 16 (2 часа)

Тема: «Технология Token Ring»

3.16.1 Задачи работы:

1. Изучить технологию Token Ring;
2. Ознакомиться с маркерным методом доступа к разделяемой среде;
3. Рассмотреть физический уровень технологии Token Ring.

3.16.2 Краткое описание практического:

Технология Token Ring

Физический уровень технологии Token Ring

3.16.3Результаты и выводы:

Сети Token Ring, так же как и сети Ethernet, характеризует разделяемая среда передачи данных, которая в данном случае состоит из отрезков кабеля, соединяющих все станции сети в кольцо. Кольцо рассматривается как общий разделяемый ресурс, и для доступа к нему требуется не случайный алгоритм, как в сетях Ethernet, а детерминированный, основанный на передаче станциям права на использование кольца в определенном порядке. Это право передается с помощью кадра специального формата, называемого *маркером* или *токеном (token)*.

Технология Token Ring был разработана компанией IBM в 1984 году, а затем передана в качестве проекта стандарта в комитет IEEE 802, который на ее основе принял в 1985 году стандарт 802.5.

Сети Token Ring работают с двумя битовыми скоростями - 4 и 16 Мбит/с. Смешение станций, работающих на различных скоростях, в одном кольце не допускается.

Технология Token Ring является более сложной технологией, чем Ethernet. Она обладает свойствами отказоустойчивости. В сети Token Ring определены процедуры контроля работы сети, которые используют обратную связь кольцеобразной структуры - посланный кадр всегда возвращается в станцию - отправитель. В некоторых случаях обнаруженные ошибки в работе сети устраняются автоматически, например может быть восстановлен потерянный маркер. В других случаях ошибки только фиксируются, а их устранение выполняется вручную обслуживающим персоналом.

Для контроля сети одна из станций выполняет роль так называемого *активного монитора*. Активный монитор выбирается во время инициализации кольца как станция с максимальным значением MAC-адреса, Если активный монитор выходит из строя, процедура инициализации кольца повторяется и выбирается новый активный монитор. Чтобы сеть могла обнаружить отказ активного монитора, последний в работоспособном состоянии каждые 3 секунды генерирует специальный кадр своего присутствия. Если этот кадр не появляется в сети более 7 секунд, то остальные станции сети начинают процедуру выборов нового активного монитора.

Маркерный метод доступа к разделяемой среде

В сетях с *маркерным методом доступа* (а к ним, кроме сетей Token Ring, относятся сети FDDI) право на доступ к среде передается циклически от станции к станции по логическому кольцу.

В сети Token Ring кольцо образуется отрезками кабеля, соединяющими соседние станции. Таким образом, каждая станция связана со своей предшествующей и последующей станцией и может непосредственно обмениваться данными только с ними. Для обеспечения доступа станций к физической среде по кольцу циркулирует кадр специального формата и назначения - маркер. В сети Token Ring любая станция всегда непосредственно получает данные только от одной станции - той, которая является предыдущей в кольце. Такая станция называется *ближайшим активным соседом, расположенным выше по потоку* (данных) - *Nearest Active Upstream Neighbor, NAUN*. Передачу же данных станция всегда осуществляет своему ближайшему соседу вниз по потоку данных.

Получив маркер, станция анализирует его и при отсутствии у нее данных для передачи обеспечивает его продвижение к следующей станции. Станция, которая имеет данные для передачи, при получении маркера изымает его из кольца, что дает ей право

доступа к физической среде и передачи своих данных. Затем эта станция выдает в кольцо кадр данных установленного формата последовательно по битам. Переданные данные проходят по кольцу всегда в одном направлении от одной станции к другой. Кадр снабжен адресом назначения и адресом источника.

Все станции кольца ретранслируют кадр побитно, как повторители. Если кадр проходит через станцию назначения, то, распознав свой адрес, эта станция копирует кадр в свой внутренний буфер и вставляет в кадр признак подтверждения приема. Станция, выдавшая кадр данных в кольцо, при обратном его получении с подтверждением приема изымает этот кадр из кольца и передает в сеть новый маркер для обеспечения возможности другим станциям сети передавать данные. Такой алгоритм доступа применяется в сетях Token Ring со скоростью работы 4 Мбит/с, описанных в стандарте 802.5.

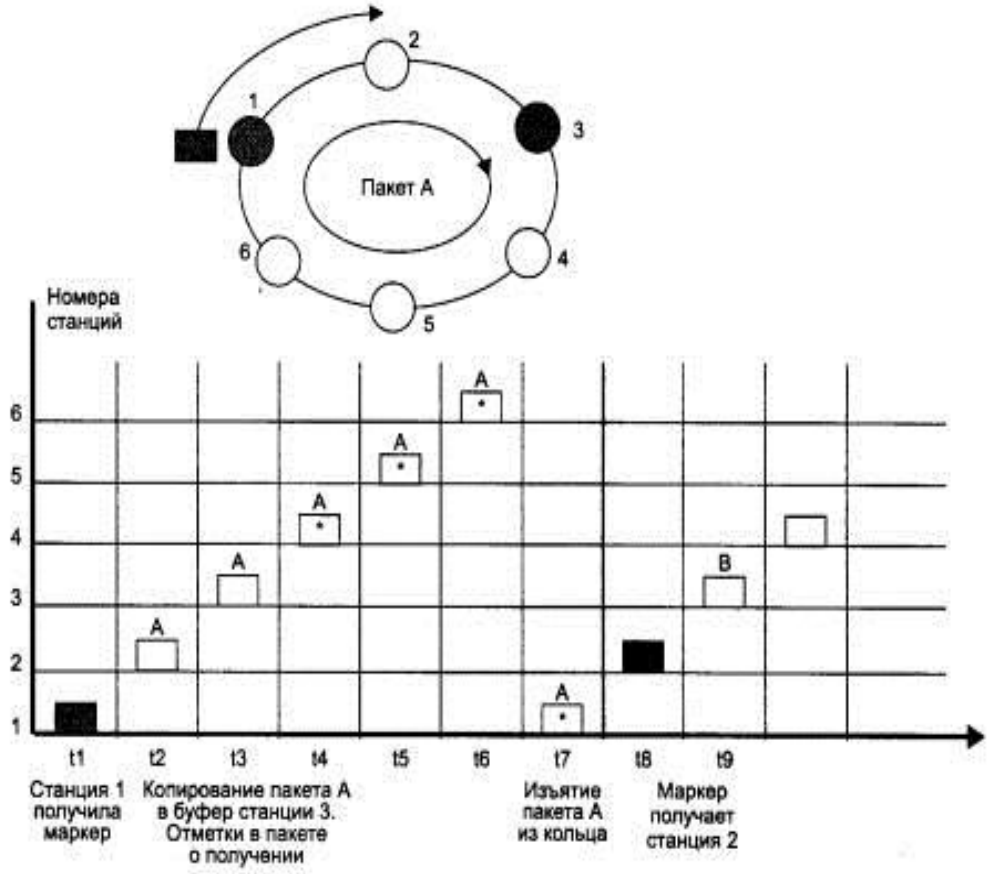


Рисунок 1 – Принцип маркерного доступа

На рисунке 1 описанный алгоритм доступа к среде иллюстрируется временной диаграммой. Здесь показана передача пакета А в кольце, состоящем из 6 станций, от станции *1* к станции *3*. После прохождения станции назначения *3* в пакете А устанавливаются два признака - признак распознавания адреса и признак копирования пакета в буфер (что на рисунке отмечено звездочкой внутри пакета). После возвращения пакета в станцию *1* отправитель распознает свой пакет по адресу источника и удаляет пакет из кольца. Установленные станцией *3* признаки говорят станции-отправителю о том, что пакет дошел до адресата и был успешно скопирован им в свой буфер.

Время владения разделяемой средой в сети Token Ring ограничивается *временем удержания маркера*, после истечения которого станция обязана прекратить передачу собственных данных (текущий кадр разрешается завершить) и передать маркер далее по кольцу. Станция может успеть передать за время удержания маркера один или несколько

кадров в зависимости от размера кадров и величины времени удержания маркера. Обычно время удержания маркера по умолчанию равно 10 мс.

Для различных видов сообщений передаваемым кадрам могут назначаться различные *приоритеты*: от 0 (низший) до 7 (высший). Решение о приоритете конкретного кадра принимает передающая станция. Маркер также всегда имеет некоторый уровень текущего приоритета. Станция имеет право захватить переданный ей маркер только в том случае, если приоритет кадра, который она хочет передать, выше (или равен) приоритета маркера. В противном случае станция обязана передать маркер следующей по кольцу станции.

За наличие в сети маркера, причем единственной его копии, отвечает активный монитор. Если активный монитор не получает маркер в течение длительного времени (например, 2,6 с), то он порождает новый маркер.

Физический уровень технологии Token Ring

Стандарт Token Ring фирмы IBM изначально предусматривал построение связей в сети с помощью концентраторов, называемых MAU (Multistation Access Unit) или MSAU (Multi-Station Access Unit), то есть устройствами многостанционного доступа (рисунок 6.3). Сеть Token Ring может включать до 260 узлов.

Концентратор Token Ring может быть активным или пассивным. Пассивный концентратор просто соединяет порты внутренними связями так, чтобы станции, подключаемые к этим портам, образовали кольцо. Такое устройство можно считать простым кроссовым блоком за одним исключением - MSAU обеспечивает обход какого-либо порта, когда присоединенный к этому порту компьютер выключают. Такая функция необходима для обеспечения связности кольца вне зависимости от состояния подключенных компьютеров.

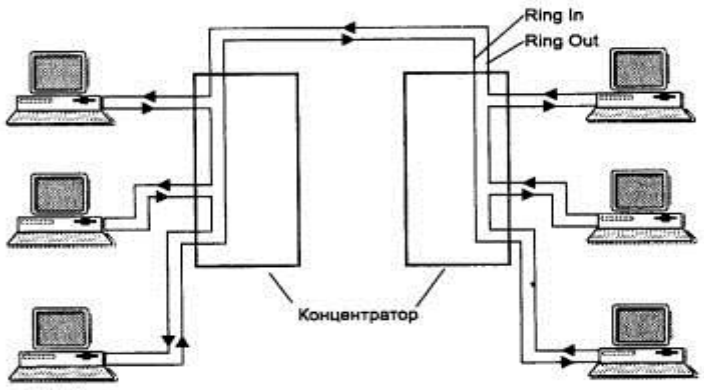


Рисунок 2 – Физическая конфигурация сети Token Ring

Активный концентратор выполняет функции регенерации сигналов и поэтому иногда называется повторителем, как в стандарте Ethernet.

Возникает вопрос – если концентратор является пассивным устройством, то каким образом обеспечивается качественная передача сигналов на большие расстояния, которые возникают при включении в сеть нескольких сот компьютеров? Ответ состоит в том, что роль усилителя сигналов в этом случае берет на себя каждый сетевой адаптер, а роль ресинхронизирующего блока выполняет сетевой адаптер активного монитора кольца. Каждый сетевой адаптер Token Ring имеет блок повторения, который умеет регенерировать и ресинхронизировать сигналы, однако последнюю функцию выполняет в кольце только блок повторения активного монитора.

Максимальная длина кольца Token Ring составляет 4000 м. Ограничения на максимальную длину кольца и количество станций в кольце в технологии Token Ring не являются такими жесткими, как в технологии Ethernet. Здесь эти ограничения во многом связаны со временем оборота маркера по кольцу (но не только – есть и другие соображения, диктующие выбор ограничений). Так, если кольцо состоит из 260 станций, то при времени удержания маркера в 10 мс маркер вернется в активный монитор в худшем случае через 2,6 с, а это время как раз составляет тайм-аут контроля оборота маркера.

3.17 Практическое занятие № 17 (2 часа)
Тема: «Архитектура FDDI»
3.17.1 Задачи работы:
1. Изучить архитектуру FDDI
3.17.2 Краткое описание практического:
Архитектура FDDI
3.17.3Результаты и выводы:

Технология FDDI (Fiber Distributed Data Interface)- оптоволоконный интерфейс распределенных данных - это первая технология локальных сетей, в которой средой передачи данных является волоконно-оптический кабель. Работы по созданию технологий и устройств для использования волоконно-оптических каналов в локальных сетях начались в 80-е годы, вскоре после начала промышленной эксплуатации подобных каналов в территориальных сетях. Проблемная группа X3T9.5 института ANSI разработала в период с 1986 по 1988 гг. начальные версии стандарта FDDI, который обеспечивает передачу кадров со скоростью 100 Мбит/с по двойному волоконно-оптическому кольцу длиной до 100 км.
Технология FDDI во многом основывается на технологии Token Ring, развивая и совершенствуя ее основные идеи. Разработчики технологии FDDI ставили перед собой в качестве наиболее приоритетных следующие цели:
повысить битовую скорость передачи данных до 100 Мбит/с;
повысить отказоустойчивость сети за счет стандартных процедур восстановления ее после отказов различного рода - повреждения кабеля, некорректной работы узла, концентратора, возникновения высокого уровня помех на линии и т. п.;
максимально эффективно использовать потенциальную пропускную способность сети как для асинхронного, так и для синхронного (чувствительного к задержкам) трафиков.
Сеть FDDI строится на основе двух оптоволоконных колец, которые образуют основной и резервный пути передачи данных между узлами сети. Наличие двух колец - это основной способ повышения отказоустойчивости в сети FDDI, и узлы, которые хотят воспользоваться этим повышенным потенциалом надежности, должны быть подключены к обоим кольцам.
В нормальном режиме работы сети данные проходят через все узлы и все участки кабеля только первичного (Primary) кольца, этот режим назван режимом Thru - «сквозным» или «транзитным». Вторичное кольцо (Secondary) в этом режиме не используется.
В случае какого-либо вида отказа, когда часть первичного кольца не может передавать данные (например, обрыв кабеля или отказ узла), первичное кольцо объединяется со вторичным (рис. 3.16), вновь образуя единое кольцо. Этот режим работы сети называется Wrap, то есть «свертывание» или «сворачивание» колец. Операция свертывания производится средствами концентраторов и/или сетевых адаптеров FDDI. Для упрощения этой процедуры данные по первичному кольцу всегда передаются в одном направлении (на диаграммах это направление изображается против часовой стрелки), а по вторичному -

в обратном (изображается по часовой стрелке). Поэтому при образовании общего кольца из двух колец передатчики станций по-прежнему остаются подключенными к приемникам соседних станций, что позволяет правильно передавать и принимать информацию соседними станциями.

В стандартах FDDI много внимания отводится различным процедурам, которые позволяют определить наличие отказа в сети, а затем произвести необходимую реконфигурацию. Сеть FDDI может полностью восстанавливать свою работоспособность в случае единичных отказов ее элементов. При множественных отказах сеть распадается на несколько не связанных сетей. Технология FDDI дополняет механизмы обнаружения отказов технологии Token Ring механизмами реконфигурации пути передачи данных в сети, основанными на наличии резервных связей, обеспечиваемых вторым кольцом.

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

Отличия метода доступа заключаются в том, что время удержания маркера в сети FDDI не является постоянной величиной, как в сети Token Ring. Это время зависит от загрузки кольца - при небольшой загрузке оно увеличивается, а при больших перегрузках может уменьшаться до нуля. Эти изменения в методе доступа касаются только асинхронного трафика, который не критичен к небольшим задержкам передачи кадров. Для синхронного трафика время удержания маркера по-прежнему остается фиксированной величиной. Механизм приоритетов кадров, аналогичный принятому в технологии Token Ring, в технологии FDDI отсутствует. Разработчики технологии решили, что деление трафика на 8 уровней приоритетов избыточно и достаточно разделить трафик на два класса - асинхронный и синхронный, последний из которых обслуживается всегда, даже при перегрузках кольца.

В остальном пересылка кадров между станциями кольца на уровне MAC полностью соответствует технологии Token Ring. Станции FDDI применяют алгоритм раннего освобождения маркера, как и сети Token Ring со скоростью 16 Мбит/с.

Адреса уровня MAC имеют стандартный для технологий IEEE 802 формат. Формат кадра FDDI близок к формату кадра Token Ring, основные отличия заключаются в отсутствии полей приоритетов. Признаки распознавания адреса, копирования кадра и ошибки позволяют сохранить имеющиеся в сетях Token Ring процедуры обработки кадров станцией-отправителем, промежуточными станциями и станцией-получателем.

На рис. 3.17 приведено соответствие структуры протоколов технологии FDDI семиуровневой модели OSI. FDDI определяет протокол физического уровня и протокол подуровня доступа к среде (MAC) канального уровня. Как и во многих других технологиях локальных сетей, в технологии FDDI используется протокол подуровня управления каналом данных LLC, определенный в стандарте IEEE 802.2. Таким образом, несмотря на то что технология FDDI была разработана и стандартизована институтом ANSI, а не комитетом IEEE, она полностью вписывается в структуру стандартов 802.

Отличительной особенностью технологии FDDI является уровень управления станцией - Station Management (SMT). Именно уровень SMT выполняет все функции по управлению и мониторингу всех остальных уровней стека протоколов FDDI. В управлении кольцом принимает участие каждый узел сети FDDI. Поэтому все узлы обмениваются специальными кадрами SMT для управления сетью.

Отказоустойчивость сетей FDDI обеспечивается протоколами и других уровней: с помощью физического уровня устраняются отказы сети по физическим причинам, например из-за обрыва кабеля, а с помощью уровня MAC - логические отказы сети, например потеря нужного внутреннего пути передачи маркера и кадров данных между портами концентратора.

3.18 Практическое занятие № 18 (2 часа)
Тема: «Архитектура АТМ»
3.18.1 Задачи работы:
1. Изучить архитектуру АТМ
3.18.2 Краткое описание практического:
Архитектура АТМ
3.18.3Результаты и выводы:

Архитектура ATM. Такие технологии передачи, как Ethernet и Token Ring, соответствуют семиуровневой модели взаимодействия открытых систем Open Systems Interconnection - OSI . ATM же имеет собственную модель, разработанную организациями по стандартизации. Технология ATM была разработана организациями ANSI и ITU как транспортный механизм для широкополосной сети ISDN Broadband Integrated Services Digital Network - B-ISDN . B-ISDN - это общедоступная территориально-распределенная сеть WAN , которая может использоваться для объединения нескольких локальных сетей.
Впоследствии ATM Forum - консорциум производителей оборудования для сетей ATM - приспособил и расширил стандарты B-ISDN для использования как в общедоступных, так и в частных сетях см. врезку Организации по стандартизации ATM . Модель ATM, в соответствии с определением ANSI, ITU и ATM Forum, состоит из трех уровней физического уровня ATM уровня адаптации ATM. Эти три уровня примерно соответствуют по функциям физическому, канальному и сетевому уровню модели OSI рисунок 1 . В настоящее время модель ATM не включает в себя никаких дополнительных уровней, т.е. таких, которые соответствуют более высоким уровням модели OSI. Однако самый высокий уровень в модели ATM может связываться непосредственно с физическим, канальным, сетевым или транспортным уровнем модели OSI, а также непосредственно с ATM-совместимым приложением. Рисунок 1. В отличие от других протоколов передачи, ATM используетсобственную модель, а не модель OSI. Физический уровень Как в модели ATM, так и в модели OSI стандарты для физического уровня устанавливают, каким образом биты должны проходить через среду передачи.
Точнее говоря, стандарты ATM для физического уровня определяют, как получать биты из среды передачи, преобразовывать их в ячейки и посылать эти ячейки уровню ATM. Стандарты ATM для физического уровня также описывают, какие кабельные системы должны использоваться в сетях ATM и с какими скоростями может работать ATM при каждом типе кабеля.
Изначально ATMForum установил скорость DS3 45 Мбит с и более высокие.
Однако реализация ATM со скоростью 45 Мбит с применяется главным образом провайдерами услуг WAN. Другие же компании чаще всего используют ATM со скоростью 25 или 155 Мбит с. Хотя ATM Forum первоначально не принял реализацию ATM со скоростью 25 Мбит с, отдельные производители стали ее сторонниками, поскольку такое оборудование дешевле в производстве и установке, чем работающее на других скоростях.
Только 25-мегабитная ATM может работать на неэкранированной витой паре UTP категории 3, а также на UTP более высокой категории и оптоволоконном кабеле. Вследствие того что оборудование для 25-мегабитной ATM относительно недорого, оно предназначено для подключения к сети ATM настольных компьютеров см. врезку Более доступный вариант ATM со скоростью 25 Мбит с . 155-мегабитная ATM работает на кабелях UTP категории 5, экранированной витой паре STP типа 1, оптоволоконном кабеле и беспроводных инфракрасных лазерных каналах. 622-мегабитная ATM работает только на оптоволоконном кабеле и может использоваться в локальных сетях хотя оборудование, работающее с такой скоростью, реализовано еще недостаточно широко. А для

беспроводной связи лаборатория Olivetti Research Labs создает прототип радиосети ATM, работающей со скоростью 10 Мбит с. Уровень ATM и виртуальные каналы В модели OSI стандарты для канального уровня описывают, каким образом устройства могут совместно использовать среду передачи и гарантировать надежное физическое соединение.
Стандарты для уровня ATM регламентируют передачу сигналов, управление трафиком и установление соединений в сети ATM. Функции передачи сигналов и управления трафиком уровня ATM подобны функциям канального уровня модели OSI, а функции установления соединения ближе всего к функциям маршрутизации, которые определены стандартами модели OSI для сетевого уровня.
Стандарты для уровня ATM описывают, как получать ячейку, сгенерированную на физическом уровне, добавлять 5-байтный заголовок и посылать ячейку уровню адаптации ATM. Эти стандарты также определяют, каким образом нужно устанавливать соединение с таким качеством сервиса QoS , которое запрашивает ATM-устройство или конечная станция. Стандарты установления соединения для уровня ATM определяют виртуальные каналы и виртуальные пути. Виртуальный канал ATM - это соединение между двумя конечными станциями ATM, которое устанавливается на время их взаимодействия.
Виртуальный канал является двунаправленным это означает, что после установления соединения каждая конечная станция может как посылать пакеты другой станции, так и получать их от нее. После того как соединение установлено, коммутаторы между конечными станциями получают адресные таблицы, содержащие сведения о том, куда необходимо направлять ячейки.
В них используется следующая информация адрес порта, из которого приходят ячейки специальные значения в заголовках ячейки, которые называются идентификаторами виртуального канала virtual circuit identifiers - VCI и идентификаторами виртуального пути virtual path identifiers - VPI . Адресные таблицы также определяют, какие VCI и VPI коммутатор должен включить в заголовки ячеек перед тем как их передать.
Имеются три типа виртуальных каналов постоянные виртуальные каналы permanent virtual circuits - PVC коммутируемые виртуальные каналы switched virtual circuits - SVC интеллектуальные постоянные виртуальные каналы smart permanent virtual circuits - SPVC . PVC - это постоянное соединение между двумя конечными станциями, которое устанавливается вручную в процессе конфигурирования сети. Пользователь сообщает провайдеру ATM-услуг или сетевому администратору, какие конечные станции должны быть соединены, и он устанавливает PVC между этими конечными станциями.
PVC включает в себя конечные станции, среду передачи и все коммутаторы, расположенные между конечными станциями.
После установки PVC для него резервируется определенная часть полосы пропускания, и двум конечным станциям не требуется устанавливать или сбрасывать соединение. SVC устанавливается по мере необходимости - всякий раз, когда конечная станция пытается передать данные другой конечной станции. Когда отправляющая станция запрашивает соединение, сеть ATM распространяет адресные таблицы и сообщает этой станции, какие VCI и VPI должны быть включены в заголовки ячеек. Через произвольный промежуток времени SVC сбрасывается.
SVC устанавливается динамически, а не вручную. Для него стандарты передачи сигналов уровня ATM определяют, как конечная станция должна устанавливать, поддерживать и сбрасывать соединение. Эти стандарты также регламентируют использование конечной станцией при установлении соединения параметров QoS из уровня адаптации ATM. Кроме того, стандарты передачи сигналов описывают способ управления трафиком и предотвращения заторов соединение устанавливается только в том случае, если сеть в состоянии поддерживать это соединение.
Процесс определения, может ли быть установлено соединение, называется управлением признанием соединения connection admission control - CAC . SPVC - это гибрид PVC и SVC. Подобно PVC, SPVC устанавливается вручную на этапе конфигурирования сети.

Однако провайдер ATM-услуг или сетевой администратор задает только конечные станции. Для каждой передачи сеть определяет, через какие коммутаторы будут передаваться ячейки. Большая часть раннего оборудования ATM поддерживала только PVC. Поддержка SVC и SPVC начинает реализовываться только сейчас.
PVC имеют два преимущества над SVC. Сеть, в которой используются SVC, должна тратить время на установление соединений, а PVC устанавливаются предварительно, поэтому могут обеспечить более высокую производительность. Кроме того, PVC обеспечивают лучший контроль над сетью, так как провайдер ATM-услуг или сетевой администратор может выбирать путь, по которому будут передаваться ячейки.
Однако и SVC имеют ряд преимуществ перед PVC. Поскольку SVC устанавливается и сбрасывается легче, чем PVC, то сети, использующие SVC, могут имитировать сети без установления соединений. Эта возможность оказывается полезной в том случае, если вы используете приложение, которое не может работать в сети с установлением соединений. Кроме того, SVC используют полосу пропускания, только когда это необходимо, а PVC должны постоянно ее резервировать на тот случай, если она понадобится.
SVC также требуют меньшей административной работы, поскольку устанавливаются автоматически, а не вручную. И наконец, SVC обеспечивают отказоустойчивость когда выходит из строя коммутатор, находящийся на пути соединения, другие коммутаторы выбирают альтернативный путь. В некотором смысле SPVC обладает лучшими свойствами этих двух видов виртуальных каналов. Как и в случае с PVC, SPVC позволяет заранее задать конечные станции, поэтому им не приходится тратить время на установление соединения каждый раз, когда одна из них должна передать ячейки.
Подобно SVC, SPVC обеспечивает отказоустойчивость. Однако и SPVC имеет свои недостатки как и PVC, SPVC устанавливается вручную, и для него необходимо резервировать часть полосы пропускания - даже если он не используется. Стандарты установления соединения для уровня ATM также определяют виртуальные пути virtual path. В то время как виртуальный канал - это соединение, установленное между двумя конечными станциями на время их взаимодействия, виртуальный путь - это путь между двумя коммутаторами, который существует постоянно, независимо от того, установлено ли соединение.
Другими словами, виртуальный путь - это запомненный путь, по которому проходит весь трафик от одного коммутатора к другому. Когда пользователь запрашивает виртуальный канал, коммутаторы определяют, какой виртуальный путь использовать для достижения конечных станций. По одному и тому же виртуальному пути в одно и то же время может передаваться трафик более чем для одного виртуального канала.
Например, виртуальный путь с полосой пропускания 120 Мбит с может быть разделен на четыре одновременных соединения по 30 Мбит с каждый. Уровень адаптации ATM и качество сервиса В модели OSI стандарты для сетевого уровня определяют, как осуществляется маршрутизация пакетов и управление ими. В модели ATM стандарты для уровня адаптации ATM выполняют три подобные функции определяют, как форматируются пакеты предоставляют информацию для уровня ATM, которая дает возможность этому уровню устанавливать соединения с различным QoS предотвращают заторы. Уровень адаптации ATM состоит из четырех протоколов называемых протоколами AAL , которые форматируют пакеты.