Белорусский национальный технический университет
Факультет Международный институт дистанционного образования
Кафедра «Информационные системы и технологии»

ЭЛЕКТРОННЫЙ УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

«КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ»

для специальности:
1- 40 01 01 «Программное обеспечение информационных технологий»

Составитель:
Русак Леонид Владимирович, старший преподаватель каф. ИСиТ

Минск БНТУ 2023

Перечень материалов

Электронный учебно-методический комплекс включает:

- теоретический раздел (конспект лекций),
- практический раздел (лабораторные занятия),
- контроль знаний (задания контрольной работе),
- вспомогательный раздел (программа дисциплины, литература, список вопросов)

Пояснительная записка

Электронный учебно-методический комплекс разработан для студентов специальности 1 - 1-40 01 01 «Программное обеспечение информационных технологий». Информационное наполнение ЭУМК соответствует программе дисциплины «Компьютерные системы и сети».

ЭУМК может использоваться как при проведении занятий по дисциплине «Компьютерные системы и сети», так и для организации самостоятельной работы студентов. Внедрение ЭУМК будет способствовать более эффективному овладению теоретическими и практическими основами построения и функционирования компьютерных сетей и сетевого программирования.

Информация в ЭУМК хорошо структурирована. Теоретический раздел включает основные темы курса. Лабораторный практикум содержит необходимый базовый методический материал (цель лабораторной работы, краткие теоретические сведения, задания на лабораторную работу, контрольные вопросы). В ЭУМК приводится также список литературы и актуальная программа дисциплины. Модуль контроля знаний состоит из экзаменационных вопросов.

ЭУМК разработан в виде pdf-файла и не требует установки специального программного обеспечения

СОДЕРЖАНИЕ

РАЗДЕЛ 1. ТЕОРЕТИЧЕСКИЙ

1. Общие принципы построения компьютерных сетей

1.1. Определение компьютерной сети. Обобщенная схема функционирования сети

Компьютерная сеть (КС) — это единый комплекс, где компьютеры, серверы и другая техника взаимодействуют посредством каналов связи. Основное назначение комплекса состоит в упрощении и облегчении IT-процессов и ускорении работы. Пользователи получают совместный доступ к аппаратному и программному обеспечению, а также информационным ресурсам.

Можно выделить 5 этапов причин и предпосылок развития компьютерных сетей:

1) ***Системы пакетной обработки***.

Сети строились на базе Мейнфрейм (большая универсальная машина, самые мощные вычислительные системы общего назначения обеспечивают непрерывный круглосуточный режим работы; могут включать один или несколько процессоров). Пользователи готовили перфокарты, содержащие данные и команды программ. Операторы вводили эти карты в Мейнфрейм, а распечатки результатов пользователи получали только на следующий день.

2) ***Многотерминальные системы.***

Каждый пользователь получал в своё распоряжение терминал (устройство ввода-вывода). Все терминалы связывались с центральным. Обработка информации была централизованной.

3) ***Появление глобальных сетей***.

Связанно с потребностью в соединении ПК, находящихся на расстоянии друг от друга с помощью телефонной сети.

4) ***Появление первых локальных сетей.***

Связанно с появлением линии ПК. Отличительная черта – использование устройств сопряжения.

5) ***Создание стандартных технологий локальных сетей.***

Связанно с появлением ПК.

В середине 80-х гг. прошлого века были утверждены стандартные технологии объединения ПК в сеть. Теперь для построения локальных сетей достаточно было подключить сетевые адаптеры соответствующего стандарта к стандартному кабелю и установить на ПК сетевую операционную систему (ОС).

Таким образом появление компьютерных сетей было вызвано потребностью обмениваться информацией между ПК.

Начальном этапе, стоит рассмотреть более подробно базовые понятия для общего понимания функционирования локальной вычислительной сети (ЛВС), в основе которой лежит понятие телекоммуникации.

Телекоммуникация (греч. tele – вдаль, далеко и лат. communicatio – общение) – передача данных на большие расстояния.

Средства телекоммуникации – совокупность технических, программных и организационных средств для передачи данных на большие расстояния.

Телекоммуникационная сеть – множество средств телекоммуникации, связанных между собой и образующих сеть определённой топологии (конфигурации). Телекоммуникационными сетями являются (рис.1.1):

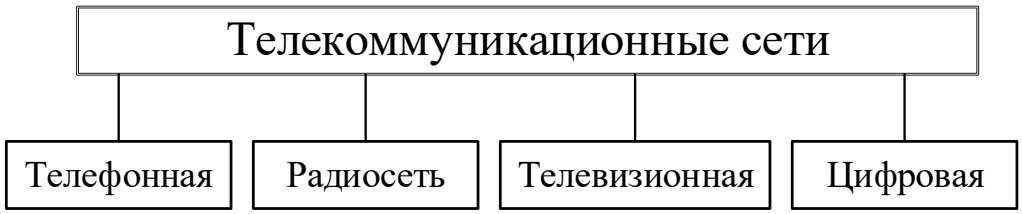


Рисунок 1.1. – Виды телекоммуникационных сетей

• телефонные сети для передачи телефонных данных (голоса);
• радиосети для передачи аудиоданных;
• телевизионные сети для передачи видеоданных;
• цифровые (компьютерные) сети или сети передачи данных (СПД);
• для передачи цифровых (компьютерных) данных.

Данные в цифровых телекоммуникационных сетях формируются в виде сообщений, имеющих определенную структуру и рассматриваемых как единое целое.

Данные (сообщения) могут быть:
• непрерывными;
• дискретными.

Непрерывные данные могут быть представлены в виде непрерывной функции времени, например, речь, звук, видео. Дискретные данные состоят из знаков (символов).

Передача данных в телекоммуникационной сети осуществляется с помощью их физического представления – сигналов.

В компьютерных сетях для передачи данных используются следующие типы сигналов (рис.1.2):

• электрический (электрический ток);

• оптический (свет);

• электромагнитный (электромагнитное поле излучения – радиоволны).

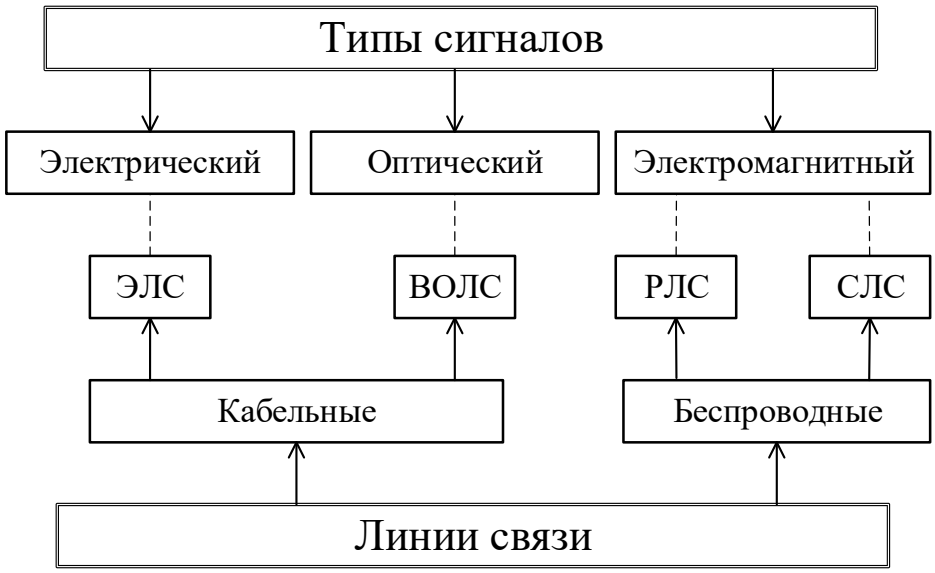


Рисунок 1.2. – Типы сигналов

Для передачи электрических и оптических сигналов применяются кабельные линии связи соответственно (рис.1.2):

• электрические (ЭЛС);

• волоконно-оптические (ВОЛС).

Передача электромагнитных сигналов осуществляется через радиолинии (РЛС) и спутниковые линии связи (СЛС).

Обратим внимание на организацию коммуникационной сети в случае рассмотрения организации передачи данных с точки зрения формирования информационных потоков.

Коммуникационная сеть - это соединение определенным образом участвующих в коммуникационном процессе участников обмена данными с помощью информационных потоков. В данном случае рассматриваются коммуникационные отношения между участниками. Коммуникационная сеть включает потоки посланий или сигналов между двумя и более участниками. Коммуникационная сеть организуется с соблюдением определенных правилах обмена данными на организации.

Можно выделить три вида коммуникационных сетей:

• открытые, в этих сетях движение команды или информации может быть остановлено, так как попадает в тупик, т.е. к элементу структуры управления, которое препятствует дальнейшей передачи данных (сигнала).

• замкнутые, в них обрыв передачи либо отсутствуют, либо может быть обойден.

• комбинированные сети сочетают в себе оба принципа построения и

присущи крупным многоуровневым предприятиям.

В случае организации более сложных сетей и делая акцент на непосредственно передачу данных, вводится определение информационных сетей.

Информационная сеть – это коммуникационная сеть, в которой информация выступает в качестве продукта создания, переработки, хранения и использования. В таких сетях осуществляются сеансы информационного взаимодействия разных категорий пользователей.

С точки зрения организации существует разделение сетей на три вида: реальные, искусственные и одноранговые.

К реальным сетям относят сети, в которых компьютеры соединяются между собой по определенной схеме посредством специальных устройств – сетевых адаптеров и требуют присутствия специалистов, осуществляющих контроль и эксплуатацию таких сетей. Их называют сетями с выделенным сервером (англ. «***real network***» или «***Network With an Attitude***», ***NWA***). Эти сети позволяют осуществлять централизованное управление. Более сложной и одновременно распространенной считается технология сети «клиент/сервер», когда любой компьютер сети в определенных ситуациях может быть попеременно как сервером, так и клиентом.

Искусственные сети не требуют специального сетевого жесткого диска. Компьютеры в этих сетях связываются между собой через последовательные или параллельные порты без специальных сетевых адаптеров. Иногда такая связь называется ноль-модемной или ноль-слотовой, так как ни в один из слотов компьютера не включена сетевая плата (адаптер).

Одноранговые сети организуются по принципу «равный среди равных» (англ. «***peer-to-peer network***»), т.е. без выделенного сервера и относятся к промежуточному типу между реальными и искусственными. В них любой компьютер попеременно может выступать в роли сервера или рабочей станции (PC)–«клиента».

Таким образом компьютерная сеть - это совокупность компьютеров, взаимо-связанных через каналы передачи данных для обеспечения обмена информацией и коллективного доступа пользователей к аппаратным, программным и информационным ресурсам сети.

В общем случае для создания компьютерных сетей требуется наличие линии связи между компьютерами (канала передачи данных), специального аппаратного обеспечения (сетевого оборудования) и специального программного обеспечения (сетевых программ-ных средств).

Компьютерные сети предназначены:

• для обмена данными между ПК;

• для совместного использования вычислительных ресурсов. Ресурсы бывают трех типов:

• аппаратные (принтер, емкости жестких дисков);

• программные;

• информационные.

1.2. Классификация, характеристики компьютерных сетей

Различают следующие виды компьютерных сетей:

• ***локальные***, которые используются для обмена информацией между ПК, расположенными на ограниченной территории (удаленными на небольшие расстояния) в пределах одного здания или в рамках одной организации;

• ***корпоративные сети*** - объединение локальных сетей в пределах одной корпорации;

• ***региональные***, связывающие компьютеры отдельной страны или экономического региона;

• ***глобальные***, используемые для связи компьютеров разных стран, континентов.

Основным требованием локальной сети является обеспечение высокой скорости обмена данными, т.е. обеспечения надежности, так как исправление возникших ошибок может свести на нет весь выигрыш в скорости передачи. Типичное количество ошибок в локальных сетях составляет один ошибочный бит на сто миллионов (то есть вероятность ошибки 10-8). Это достигается, в частности, использованием высококачественных кабелей (чаще всего это коаксиальные электрические кабели и в последнее время — оптоволоконные). Также в локальных сетях уделяется большое внимание снижению времени ожидания установления связи, так как, с точки зрения пользователя, оно входит во время передачи информации.

Таким образом, локальные сети, исходя из специфики их применения, требуют использования специальных технических и программных средств, которые должны обеспечивать быструю и удобную связь между всеми компьютерами.

Главной особенностью корпоративных сетей является их масштабность. Число пользователей и компьютеров в корпоративной сети может измеряться тысячами, а число серверов – сотнями; расстояния между сетями отдельных территорий могут оказаться такими, что использование глобальных связей становится необходимым. Непременным атрибутом корпоративной сети является высокая степень неоднородности (гетерогенности) – нельзя удовлетворить потребности тысяч пользователей с помощью однотипных

программных и аппаратных средств. В корпоративной сети обязательно используются различные типы компьютеров – от мэйнфреймов до персональных компьютеров, несколько типов операционных систем и множество различных приложений.

При организации региональных сетей сталкиваются с проблемой значительно удаленных друг от друга компьютеров (например, расположенных в разных концах города или в разных городах), между которыми необходимо организовать постоянный обмен большими потоками информации. Решение проблемы, это создание специальные постоянно действующие выделенные каналы. Физически выделенные каналы могут реализовываться с помощью телефонных каналов или оптических кабелей, а также с помощью спутниковых или радиоканалов. С помощью выделенных каналов обычно соединяются удаленные друг от друга компьютеры одной организации (например, компьютеры центрального офиса банка с компьютерами в его филиалах). Сети, связывающие значительно удаленные друг от друга компьютеры, называются распределенными. Доступ к распределенным сетям организаций ограничен определенным кругом лиц, для которых работа в таких сетях связана с выполнением их должностных обязанностей. По своему функциональному назначению сети подобного типа эквивалентны локальным и называются региональными. скорость, а также возможность передачи на большое расстояние.

Особенностями глобальных сетей является следующие.

1. Неограниченный территориальный охват.

2. Сеть объединяет ЭВМ самых разных классов (от персональных до суперЭВМ), локальные и территориальные сети разных технологий.

3. Для объединения различных сетей и передачи данных на большие расстояния используется специальное оборудование, а именно: аппаратура передачи данных (модемы, приемопередатчики и т.п.) и активное сетевое оборудование (маршрутизаторы, коммутаторы, шлюзы).

4. Топология глобальных сетей, в общем случае, произвольная.

5. Одной из важнейших задач, решаемой при построении глобальной сети, является организация эффективной маршрутизации передаваемых данных.

6. Глобальная сеть может содержать каналы связи разных типов: кабельные оптические и электрические, в том числе телефонные, беспроводные радио и спутниковые каналы, имеющие различные пропускные способности (от нескольких кбит/с до сотен Гбит/с).

Объединение глобальных, региональных и локальных вычислительных сетей позволяет создавать многосетевые иерархии. Они обеспечивают

мощные, экономически целесообразные средства обработки огромных информационных массивов и доступ к неограниченным информационным ресурсам. На рис. 1.3 приведена одна из возможных иерархий вычислительных сетей.

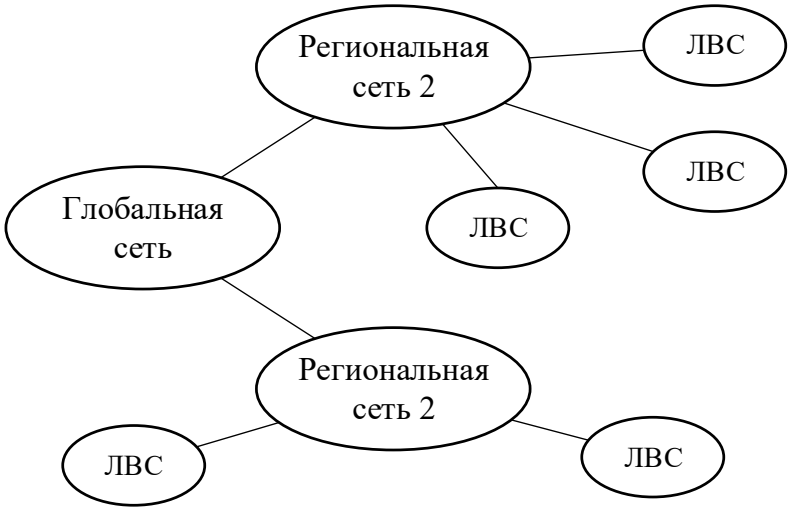


Рисунок 1.3. - Иерархия компьютерных сетей

Локальные вычислительные сети могут входить как компоненты в состав региональной сети, региональные сети — объединяться в составе глобальной сети и, наконец, глобальные сети могут также образовывать сложные структуры.

Компьютерная сеть Интернет является наиболее популярной глобальной сетью. В ее состав входит множество свободно соединенных сетей. Внутри каждой сети, входящей в Интернет, существуют конкретная структура связи и определенная дисциплина управления. Внутри Интернет структура и методы соединений между различными сетями для конкретного пользователя не имеют никакого значения.

Стоит отметить присущую разность эксплуатируемых сетей, так каждым годом усиливается тенденция сближения компьютерных и телекоммуникационных сетей разных видов, что вызывает ряд трудностей в организации высокоскоростной передачи данных с обеспечением высокого уровня надежности. Это приводит к ***конвергенции сетей***, что означает конвергенцию технологий, которая определяет возможность конвергенции различных сетевых услуг. Предпринимаются попытки создания универсальной, так называемой мультисервисной сети, способной предоставлять услуги как компьютерных, так и телекоммуникационных сетей.

К телекоммуникационным сетям относятся телефонные сети, радиосети и телевизионные сети. Главное, что объединяет их с компьютерными сетями, — то, что в качестве ресурса, предоставляемого клиентам, выступает

информация. Однако эти сети, как правило, представляют информацию в разном виде. Так, изначально компьютерные сети разрабатывались для передачи алфавитно-цифровой информации, которую часто называют просто данными, в результате у компьютерных сетей имеется и другое название — сети передачи данных, в то время как телефонные сети и радиосети были созданы для передачи только голосовой информации, а телевизионные сети передают и голос, и изображение.

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

Прежде всего, наблюдается сближение видов услуг, предоставляемых клиентам. Первая и не очень успешная попытка создания мультисервисной сети, способной оказывать различные услуги, в том числе услуги телефонии и передачи данных, привела к появлению технологии цифровых сетей с интегрированным обслуживанием (Integrated Services Digital Network, ISDN). Наибольшую привлекательность сейчас представляют собой новые виды комбинированных услуг, в которых сочетаются несколько традиционных услуг, например, услуга универсальной службы сообщений, объединяю-щей электронную почту, телефонию, факсимильную службу. Наибольших успехов на практическом поприще достигла IP-телефония, услугами которой прямо или косвенно сегодня пользуются миллионы людей.

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

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

Сегодня становится все более очевидным, что мультисервисная сеть нового поколения не может быть создана в результате «победы» какой-нибудь

одной технологии или подхода. Ее может породить только процесс конвергенции, когда от каждой технологии будет взято самое лучшее и соединено в некоторый новый сплав, который и даст требуемое качество для поддержки существующих и создания новых услуг. Появился новый термин — инфокоммуникационная сеть, который прямо говорит о двух составляющих современной сети — информационной (компьютерной) и телекоммуникационной. Новый термин еще не приобрел достаточной популярности и имеет пока достаточно узкое распространение.

1.3. Понятие протокола и применение сетевых протоколов для взаимодействия объектов сети

Со временем основной целью компьютерных развития сетей (помимо передачи информации) стала цель распределенного использования информационных ресурсов:

1. Периферийных устройств: принтеры, сканеры и т. д.
2. Данных хранящихся в оперативной памяти устройств.
3. Вычислительных мощностей.

Достичь эту цель помогали сетевые интерфейсы. Сетевые интерфейсы это определенная логическая и/или физическая граница между взаимодействующими независимыми объектами.

Сетевые интерфейсы разделяются на:

1. Физические интерфейсы (порты).
2. Логические интерфейсы (протоколы).

Порт число записывается в заголовках протоколов транспортного уровня (об этом ниже). Порт указывает для какой программы предназначен тот или иной пакет (грубо говоря та или иная информация). Например, http-сервер работает через порт 80. Когда осуществляется открываете браузера, отправляется запрос на веб-сервер через 80 порт и сервер понимая этот http запрос передает страницу в формате html (ответ сервера).

Протокол, например TCP/IP это адрес узла (компьютера) с указанием порта и передаваемых данных. Например, что бы передать информацию по протоколу TCP/IP нужно указать следующие данные:

Адрес отправителя (Source address):
IP: 82.146.49.11
Port: 2049
Адрес получателя (Destination address):
IP: 192.168.100.111
Port: 53

Данные пакета:

…

Благодаря этим данным информация будет передана на нужный узел.

В данный момент времени наиболее частая организации в передаче данных строится по технологии клиент—сервер.

Клиент — это модуль, предназначенный для формирования и передачи сообщений-запросов к ресурсам удаленного компьютера от разных приложений с последующим приемом результатов из сети и передачей их соответствующим приложениям.

Сервер — это модуль, который постоянно, ожидает прихода из сети запросов от клиентов и, приняв запрос, пытается его обслужить, как правило, с участием локальной ОС; один сервер может обслуживать запросы сразу нескольких клиентов (поочередно или одновременно)/

Другими словами, сервер — это компьютер на котором установлена программа, или принтер, клиент — это компьютер который подключается к программе, работает с ней и распечатывает какие-либо результаты, например. При этом программа может быть установлена на Клиенте, а база данных программы на Сервере.

Множество всех адресов, которые являются допустимыми в рамках некоторой схемы адресации, называется ***адресным пространством***. Адресное пространство может иметь плоскую (линейную) организацию или иерархическую организацию.

Для преобразования адресов из одного вида в другой используются специальные вспомогательные протоколы, которые называют протоколами разрешения адресов.

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

(OSI) Open System Interconnection — многоуровневая модель взаимодействия открытых систем, состоящая из семи уровней. Каждый из семи уровней предназначен для выполнения одного из этапов связи.

Для упрощения структуры большинство сетей организуются в наборы уровней, каждый последующий возводится над предыдущим.

Целью каждого уровня является предоставление неких сервисов для вышестоящих уровней. При этом от них скрываются детали реализации предоставляемого сервиса.

Протокол – формализованное правило, определяющие последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на одном уровне, но в разных узлах.

Протоколы, реализующие модель OSI носят информативный характер, имена и номера уровней используются для ознакомительных целей.

7	Прикладной уровень (application layer)
6	Представительский уровень (presentation layer)
5	Сеансовый уровень (session layer)
4	Транспортный уровень (transport layer)
3	Сетевой уровень (network layer)
2	Канальный уровень (data link layer)
1	Физический уровень (physical layer)

Рисунок 1.4. – уровни модели взаимодействия открытых систем

Физический уровень. Этот уровень определяет механические, электрические, процедурные и функциональные характеристики установления, поддержания и размыкания физического соединения между конечными системами.

Канальный уровень. Канальный уровень (уровень звена данных, информационно-канальный уровень) отвечает за надежную передачу данных через физический канал.

Сетевой уровень. Этот уровень обеспечивает возможность соединения и выбор маршрута между двумя конечными системами, подключенными к разным подсетям (сегментам), которые могут быть разделены множеством подсетей и могут находиться в разных географических пунктах.

Транспортный уровень. Обеспечивает интерфейс между процессами и сетью, устанавливает логические каналы между процессами и обеспечивает передачу по этим каналам информационных блоков.

Сеансовый уровень. Реализует установление, поддержку и завершение сеанса взаимодействия между прикладными процессами абонентов. Производит отслеживание: в какой момент и куда передает информация. На этом уровне происходит синхронизация передачи данных.

Представительский уровень. Представительный уровень (уровень представления данных) определяет синтаксис, форматы и структуры представления передаваемых данных (но не затрагивает семантику, значение данных).

Прикладной уровень. Обеспечивает непосредственную поддержку прикладных процессов и программ конечного пользователя (СУБД, текстовых процессоров, программ банковских терминалов и т.д.) и управление взаимодействием этих программ с сетью передачи данных.

1.4. Требования, предъявляемые к современным сетям

Основные требования, предъявляемые к вычислительным сетям — производительность, надежность, совместимость, управляемость, защищенность, расширяемость и масштабируемость. Наиболее важными из которых являются — производительность и надежность.

Производительность. Существует несколько основных характеристик производительности сети:

- время реакции;
- пропускная способность;
- задержка передачи и вариация задержки передачи.

Время реакции сети является интегральной характеристикой производительности сети и определяется как интервал времени между возникновением запроса к какой-либо сетевой службе и получением на него ответа.

Пропускная способность отражает объем данных, переданных сетью или ее час-тью в единицу времени. Она измеряется либо в битах в секунду, либо в пакетах в секунду. Пропускная способность может быть мгновенной, максимальной и средней.

Средняя пропускная способность вычисляется путем деления общего объема переданных данных на время их передачи, причем выбирается достаточно длительный промежуток времени — час, день или неделя.

Мгновенная пропускная способность отличается от средней тем, что для ус-реднения выбирается очень маленький промежуток времени — например, 10 мс, или 1 с.

Максимальная пропускная способность — это наибольшая мгновенная пропускная способность, зафиксированная в течение периода наблюдения.

Одной из первоначальных целей создания распределенных систем, к которым относятся и вычислительные сети, являлось достижение большей ***надежности*** по сравнению с отдельными вычислительными машинами.

Готовность или ***коэффициент готовности*** (***availability***) означает долю времени, в течение которого система может быть использована. Готовность может быть улучшена введением избыточности в структуру системы: ключевые элементы си-стемы должны существовать в нескольких экземплярах, чтобы при отказе одного из них функционирование системы обеспечивали другие.

Чтобы систему можно было отнести к высоконадежным, она должна обеспечить сохранность данных и защиту их от искажений. Кроме этого, должна поддерживаться ***согласованность*** (непротиворечивость) данных,

например, если для повышения надежности на нескольких файловых серверах хранится несколько копий данных, то нужно постоянно обеспечивать их идентичность.

Другой характеристикой надежности является ***вероятность доставки i пакета*** узлу назначения без искажений. Наряду с этой характеристикой могут использоваться и другие показатели: вероятность потери пакета, вероятность искажения отдельного бита передаваемых данных, отношение потерянных пакетов к доставленным.

Другим аспектом общей надежности является ***безопасность (security)***, то есть способность системы защитить данные от несанкционированного доступа.

Также характеристикой надежности является ***отказоустойчивость (fault tolerance)***. В сетях под отказоустойчивостью понимается способность системы скрыть от пользователя отказ отдельных ее элементов. В отказоустойчивой системе отказ одного из ее элементов приводит к некоторому снижению качества ее работы (деградации), а не к полному останову.

Расширяемость (extensibility) означает возможность сравнительно легкого до-бавления отдельных элементов сети (пользователей, компьютеров, приложений, служб), наращивания длины сегментов сети и замены существующей аппаратуры более мощной.

Масштабируемость (scalability) означает, что сеть позволяет наращивать ко-личество узлов и протяженность связей в очень широких пределах, при этом про-изводительность сети не ухудшается. Для обеспечения масштабируемости сети приходится применять дополнительное коммуникационное оборудование и спе-циальным образом структурировать сеть.

Прозрачность (transparency) сети достигается в том случае, когда сеть представля-ется пользователям не как множество отдельных компьютеров, связанных между собой сложной системой кабелей, а как единая традиционная вычислительная машина с системой разделения времени.

Управляемость сети подразумевает возможность централизованно контролировать состояние основных элементов сети, выявлять и разрешать проблемы, возникающие при работе сети, выполнять анализ производительности и планировать развитие сети.

Совместимость или интегрируемость означает, что сеть способна включать в себя самое разнообразное программное и аппаратное обеспечение, то есть в ней могут сосуществовать различные операционные системы, поддерживающие разные стеки коммуникационных протоколов, и работать

аппаратные средства и приложения от разных производителей. Сеть, состоящая из разнотипных элементов, называется неоднородной или гетерогенной, а если гетерогенная сеть работает без проблем, то она является интегрированной. Основной путь построения интегрированных сетей — использование модулей, выполненных в соответствии с открытыми стандартами и спецификациями.

Развитие компьютерной техники с последующим быстрым распространением компьютерной сети и подключение все большего количества пользователей выявило ряд проблем в организации передачи данных, так можно выделить следующий общие проблемы:

Снижение производительности. Часто приходиться сталкиваться с потерей целостности данных и скорости сети, что, как правило, связано с плохой передачей и также известно, как снижение производительности. Все сети, которые могут быть большими или маленькими, имеют проблемы с производительностью, но в больших сетях эта проблема снижения производительности высока, поскольку связь должна устанавливаться с большей областью, а также с помощью многих сетевых устройств.

Проблемы безопасности. Проблема безопасности одна из главных проблем компьютерной сети и большая проблема для инженеров по сетевой безопасности, которая обычно включает в себя защиту сети от различных кибератак, предотвращение несанкционированного доступа пользователей к системе и доступа к ней, а также поддержание целостности сети.

Идентификация хоста. Небольшие сети можно легко настроить с помощью ручной адресации, но это становится серьезной проблемой в крупных сетях, когда дело доходит до идентификации хоста. Потому что без какого-либо правильного адреса сети становится трудно установить связь в сети. Таким образом, правильная идентификация хоста необходима для сетевого взаимодействия.

Конфликты конфигурации. В основном крупным сетям приходится иметь дело с конфликтами конфигурации и загруженными сетями, поскольку через них проходит гораздо больше трафика. Но в небольших сетях доступно несколько тысяч IP-адресов с уникальными именами хостов, поэтому вероятность конфликта между устройствами меньше. Но в наши дни эта проблема стала меньше, поскольку сетевые структуры спроектированы таким образом, чтобы справляться с конфликтами конфигурации.

Проблема с производительностью. В настоящее время очень велик дневной объем данных, полученных из различных источников. Таким образом, пропускная способность сети также должна расти с учетом этого. Современные тенденции показывают постоянную необходимость увеличения

обработки большего количества данных, что с постоянной доработкой технологий в организации непосредственно самой передачи данных может повлечь возникновение проблем в пропускной способности.

Медленное подключение. Выполнение элементарной задачи по средством сети может занять много времени. Это часто вызвано передачей больших файлов на большой площади по сети. Это становится нежелательной проблемой для пользователей, при работе в компьютерной сети.

Мониторинг и техническое обслуживание. Мониторинг и обслуживание глобальной сети является одной из больших проблем текущего времени. Становится очень сложно отслеживать объем трафика в большой сети.

2. Локальные компьютерные сети

2.1. Классификация, локальных сетей

По административным отношениям между узлами можно выделить локальные сети с централизованным управлением или с выделенными серверами (серверные сети) и сети без централизованного управления или без выделенного сервера (децентрализованные), так называемые, одноранговые (одноуровневые) сети.

Локальные сети с централизованным управлением называются ***иерархическими***, а децентрализованные локальные сети ***равноправными (рабочая группа).*** Пользователи рабочей группы самостоятельно решают, какие ресурсы (в первую очередь файловые) на своем компьютере сделать общедоступными по сети. Децентрализованное управление ресурсами требует от пользователей повышенного уровня компьютерной грамотности, чтобы работать и как пользователю, и как администратору своего компьютера.

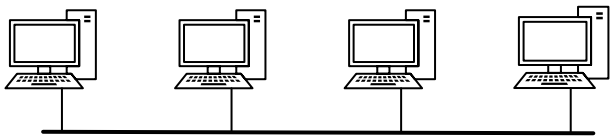


Рисунок 2.1. – Децентрализованная локальная сеть

В локальных сетях с централизованным управлением один из компьютеров является сервером, а остальные ПК - рабочими станциями.

Серверы - это высокопроизводительные компьютеры с винчестерами большой емкости и с высокоскоростной сетевой картой, которые отвечают за хранение данных, организацию доступа к этим данным и передачу данных рабочим станциям или клиентам.

Рабочие станции. Компьютеры, с которых осуществляется доступ к информации на сервере, называются рабочими станциями или клиентами.

В локальных сетях с централизованным управлением (рис 2.2) сервер обеспечивает взаимодействия между рабочими станциями, выполняет функции хранения данных общего пользования, организует доступ к этим данным и передает данные клиенту. Клиент обрабатывает полученные данные и предоставляет результаты обработки пользователю. Необходимо отметить, что обработка данных может осуществляться и на сервере.

Локальные сети с централизованным управлением, в которых сервер предназначен только хранения и выдачи клиентам информации по запросам,

называются сетями с выделенным ***файл-сервером***. Системы, в которых на сервере наряду с хранением осуществляется и обработка информации, называются системами "***клиент-сервер***".

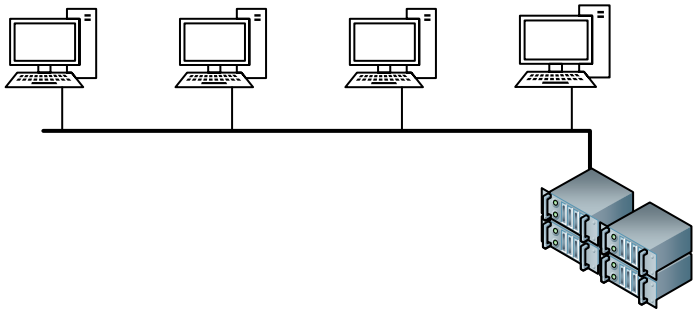


Рисунок 2.2. – Локальная сеть с централизованным управлением

Необходимо отметить, что в серверных локальных сетях клиенту непосредственно доступны только ресурсы серверов. Но рабочие станции, входящие в ЛВС с централизованным управлением, могут одновременно организовать между собой ***одноранговуюлокальную*** сеть со всеми ее возможностями.

Программное обеспечение, управляющее работой ЛВС с централизованным управлением, состоит из двух частей:

1) Сетевой операционной системы, устанавливаемой на сервере;

2) Программного обеспечения на рабочей станции, представляющего набор программ, работающих под управлением операционной системы, которая установлена на рабочей станции. При этом на разных рабочих станциях в одной сети могут быть установлены различные операционные системы.

В больших иерархических локальных сетях в качестве сетевых ОС используются UNIX и LINUX, которые являются более надежными. Для локальных сетей среднего масштаба наиболее популярной сетевой ОС является Windows 2012 Server или Windows 2016 Server.

В зависимости от способов использования сервера в иерархических сетях различают серверы следующих типов:

1) ***Файловый сервер***. В этом случае на сервере находятся совместно обрабатываемые файлы или (и) совместно используемые программы.

2) ***Сервер баз данных***. На сервере размещается сетевая база данных.

3) ***Принт-сервер***. К компьютеру подключается достаточно производительный принтер, на котором может быть распечатана информация сразу с нескольких рабочих станций.

4) ***Почтовый сервер***. На сервере хранится информация, отправляемая и

получаемая как по локальной сети.

Достоинства:

1) Выше скорость обработки данных;

2) Обладает надежной системой защиты информации и обеспечения секретности;

3) Проще в управлении по сравнению с одноранговыми сетями.

Недостатки:

1) Сеть дороже из-за выделенного сервера;

2) Менее гибкая по сравнению с равноправной сетью.

Стоит отметить, что в современной реальности возможно встретить достаточно крупные децентрализованные одноранговые сет, насчитывающих сотни и тысячи компьютеров, это приводит к сложностям в управлении ими. Данная проблема отчасти решается добавлением координационного сервера в структуру сети, такие сети называются ***гибридными***. На сервер возлагаются задачи контроля за состоянием сети, представления списка доступных ресурсов и общего управления. Например, клиенты могут обращаться к такому серверу для авторизации, после чего способны взаимодействовать друг с другом непосредственно.

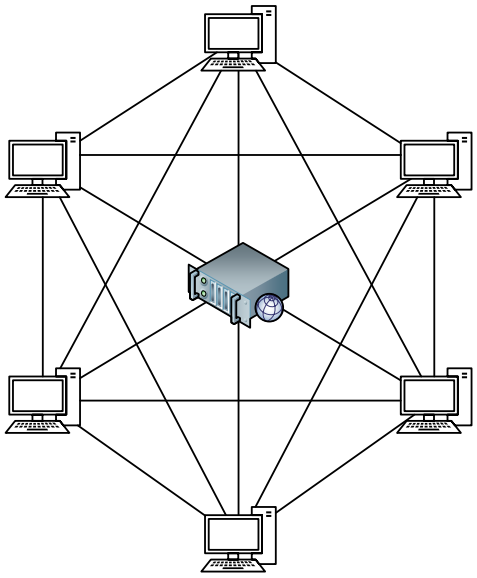


Рисунок 2.3. – Гибридная локальная сеть

В рамках одной локальной сети могут использоваться несколько выделенных серверов. По своему функциональному назначению различают несколько типов серверов: файловый, печати, коммуникационный, приложений, базы данных и т.д.

Файловый сервер — это компьютер, который выполняет функции управления локальной сетью, отвечает за коммуникационные связи, хранит

файлы, разделяемые в сети, и предоставляет доступ к совместно используемому дисковому пространству.

Сервер печати — это компьютер, программа или специальное устройство, обеспечивающее доступ станциям сети к центральному разделяемому принтеру. Запросы на печать поступают от каждой рабочей станции к серверу печати, который разделяет их на индивидуальные задания принтеру, создает очередь печати. Задания обычно обрабатываются в порядке их поступления. В функции сервера печати входит также управление принтером.

Коммуникационный сервер(сервер удаленного доступа — Access Server) позволяет работать с различными протоколами (правилами передачи информации в сети) и дает возможность станциям разделять модем или узел связи с большой ЭВМ. Это обеспечивает получение информации, хранящейся в сети, практически с любого места, где есть телефон, модем или компьютер.

Довольно часто сервер совмещает функции коммуникационного сервера и сервера приложений.

Сервер приложений выполняет одну или несколько прикладных задач, которые запускают пользователи со своих терминалов, включенных в данную сеть. Принцип действия сервера приложений такой же, как у многотерминальной системы (системы совместной обработки). Задача пользователя выполняется непосредственно на сервере приложений, по низкоскоростной телефонной линии на удаленный компьютер (терминал) передается только изображение экрана терминала пользователя, а обратно — только информация о нажимаемых пользователем клавишах. Поэтому нагрузка по передаче информации (например, при работе с базами данных) ложится на высокоскоростной кабель сети, к которой подключен сервер приложений.

2.2. Топологии локальных сетей: Физическая и логическая. Достоинства и недостатки. Выбор топологии

Рассмотрим топологию физических сетей. Сетевая топология (от греч., - место) — способ описания конфигурации сети, схема расположения и соединения сетевых устройств.

Топология – это схема соединения каналами связи компьютеров или узлов сети между собой.

Сетевая топология может быть

• физической — описывает реальное расположение и связи между узлами сети;

• логической — описывает хождение сигнала в рамках физической топологии;

• информационной — описывает направление потоков информации, передаваемых по сети;

• управления обменом — это принцип передачи права на пользование сетью.

Существует множество способов соединения сетевых устройств. Выделяют следующие топологии:

• полносвязная;

• ячеистая;

• общая шина;

• звезда;

• кольцо;

• древовидное.

1) ***Полносвязная топология*** — топология компьютерной сети, в которой каждая рабочая станция подключена ко всем остальным. Этот вариант является громоздким и неэффективным, несмотря на свою логическую простоту. Для каждой пары должна быть выделена независимая линия, каждый компьютер должен иметь столько коммуникационных портов сколько компьютеров в сети. По этим причинам сеть может иметь только сравнительно небольшие конечные размеры. Чаще всего эта топология используется в многомашинных комплексах или глобальных сетях при малом количестве рабочих станций.

Технология доступа в сетях этой топологии реализуется методом передачи маркера. Маркер – это пакет, снабженный специальной последовательностью бит (его можно сравнить с конвертом для письма). Он последовательно предается по кольцу от компьютера к компьютеру в одном направлении. Каждый узел ретранслирует передаваемый маркер. Компьютер может передать свои данные, если он получил пустой маркер. Маркер с пакетом передается, пока не обнаружится компьютер, которому предназначен пакет. В этом компьютере данные принимаются, но маркер движется дальше и возвращается к отправителю.

После того, как отправивший пакет компьютер убедится, что пакет доставлен адресату, маркер освобождается.

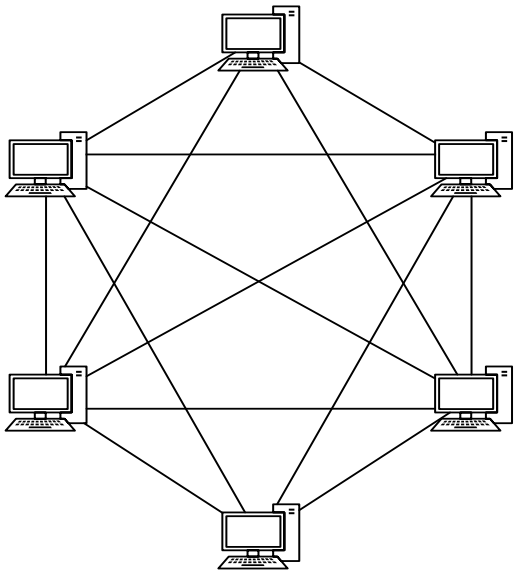


Рисунок 2.4. - Полносвязная топология

2) ***Ячеистая топология*** - базовая полносвязная топология компьютерной сети, в которой каждая рабочая станция сети соединяется с несколькими другими рабочими станциями этой же сети. Характеризуется высокой отказоустойчивостью, сложностью настройки и переизбыточным расходом кабеля. Каждый компьютер имеет множество возможных путей соединения с другими компьютерами. Обрыв кабеля не приведёт к потере соединения между двумя компьютерами.

Получается из полносвязной путем удаления некоторых возможных связей. Эта топология допускает соединение большого количества компьютеров и характерна, как правило, для крупных сете

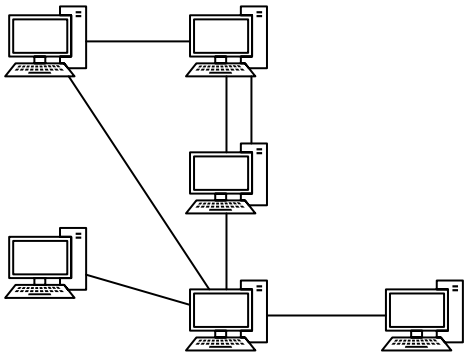


Рисунок 2.5. - Ячеистая топология

Общая шина, представляет собой общий кабель (называемый шина или магистраль), к которому подсоединены все рабочие станции. На концах кабеля

находятся терминаторы, для предотвращения отражения сигнала.

Достоинства:

- Небольшое время установки сети;
- Дешевизна (требуется меньше кабеля и сетевых устройств);
- Простота настройки;
- Выход из строя рабочей станции не отражается на работе сети.

Недостатки:

- Неполадки в сети, такие как обрыв кабеля и выход из строя терминатора, полностью блокируют работу всей сети;
- Сложная локализация неисправностей;
- С добавлением новых рабочих станций падает производительность сети.

3) ***Шинная топология*** представляет собой топологию, в которой все устройства локальной сети подключаются к линейной сетевой среде передачи данных. Такую линейную среду часто называют каналом, шиной или трассой. Каждое устройство, например, рабочая станция или сервер, независимо подключается к общему шинному кабелю с помощью специального разъема. Шинный кабель должен иметь на конце согласующий резистор, или терминатор, который поглощает электрический сигнал, не давая ему отражаться и двигаться в обратном направлении по шине.

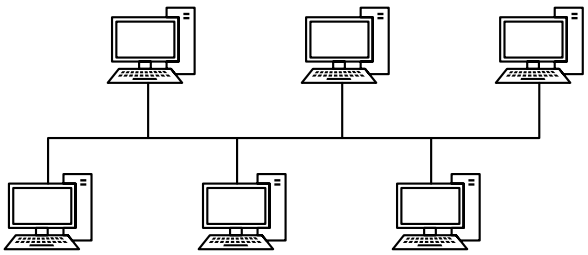


Рисунок 2.6. - Шинная топология

Достоинства:

- Простота реализации;
- Отсутствие сложной топологической настройки;
- Экономическая целесообразность.

Недостатки:

- Наличие единой точку отказа.
- Высокий сетевой трафик в следствии эксплуатации одной линии магистрали, что приводит к снижению производительности.

Это ограничение делает топологии шины подходящими только для небольших сетей. Основная причина в том, что чем больше узлов, тем ниже будет скорость передачи. Стоит также отметить, что шинные топологии

ограничены в том смысле, что они ***полудуплекс***, это означает, что данные не могут быть переданы в двух противоположных направлениях одновременно.

4)***Топология звезды*** - это топология, в которой каждый узел в сети подключен к одному центральному коммутатору. Каждое устройство в сети напрямую связано с коммутатором и косвенно связано с любым другим узлом. Связь между этими элементами заключается в том, что центральное сетевое устройство является сервером, а другие устройства рассматриваются как клиенты. Центральный узел отвечает за управление передачей данных по сети и действует как ретранслятор. В топологии «звезда» компьютеры подключаются с помощью коаксиального кабеля, витой пары или оптоволоконного кабеля.

Достоинства:

- Высокий уровень защиты от сбоев оборудования
- Быстрая замена оборудования или добавление новых участников передачи сети без потери работоспособности.
- Простота в настройке и управлении в долгосрочной перспективе.
- Простота общего дизайна.

Недостатки:

- Высокая зависимость от центрального оборудования (коммутатор)
- Производительность сети также привязаны к конфигурации и производительности центрального узла.
- Топологией Star легко управлять, но установка центрального узла далеко не дешевы.

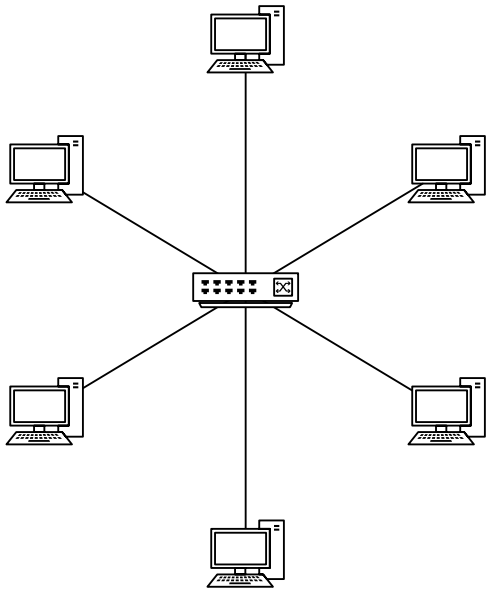


Рисунок 2.7. – Топология звезды

4) ***Кольцевая топология*** - это топология, в которой каждый компьютер соединен линиями связи только с двумя другими: от одного он только получает информацию, а другому только передает. На каждой линии связи, как и в случае звезды, работает только один передатчик и один приемник. Это позволяет отказаться от применения внешних терминаторов.

Работа в сети кольца заключается в том, что каждый компьютер ретранслирует (возобновляет) сигнал, то есть выступает в роли повторителя, потому затухание сигнала во всем кольце не имеет никакого значения, важно только затухание между соседними компьютерами кольца. Четко выделенного центра в этом случае нет, все компьютеры могут быть одинаковыми. Однако достаточно часто в кольце выделяется специальный абонент, который управляет обменом или контролирует обмен. Понятно, что наличие такого управляющего абонента снижает надежность сети, потому что выход его из строя сразу же парализует весь обмен.

Компьютеры в кольце не являются полностью равноправными (в отличие, например, от шинной топологии). Одни из них обязательно получают информацию от компьютера, который ведет передачу в этот момент, раньше, а другие — позже. Именно на этой особенности топологии и строятся методы управления обменом по сети, специально рассчитанные на «кольцо». В этих методах право на следующую передачу (или, как еще говорят, на захват сети) переходит последовательно к следующему по кругу компьютеру.

Достоинства:

- Простота установки;
- Практически полное отсутствие дополнительного оборудования;
- Возможность устойчивой работы без существенного падения скорости передачи данных при интенсивной загрузке сети, поскольку использование маркера исключает возможность возникновения коллизий.

Недостатки:

- Выход из строя одной рабочей станции, и другие неполадки (обрыв кабеля), отражаются на работоспособности всей сети;
- Сложность конфигурирования и настройки;
- Сложность поиска неисправностей.
- Необходимость иметь две сетевые платы, на каждой рабочей станции.

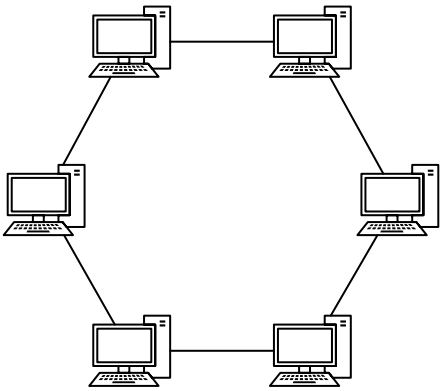


Рисунок 2.7. – Топология кольцо.

5) ***Древовидная (Иерархическая Звезда или снежинка)*** - топология типа звезды, но используется несколько концентратов, иерархически соединенных между собой связями типа звезда. Топология требует меньшей длины кабеля, чем "звезда", но больше элементов.

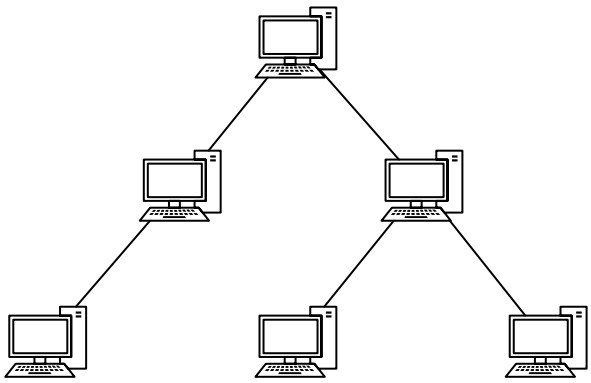


Рисунок 2.8. – Древовидная топология.

Достоинства:

- Простота в добавлении большего количества узлов в сеть;
- Простота в поиске ошибок и устранения неполадок.

Недостатки:

- Зависимость от сбоя корневого узла;
- Сложность в управлении сетью;
- Большое количество необходимых кабелей.

Топология сети, которая реализуется для предприятия, должна основываться на требованиях к использованию. Количество узлов в сети будет определять, возможно ли сделать это с помощью топологии шины или понадобится развернуть более сложную сеть или гибридную установку.

2.3. Среда передачи: проводная и беспроводная. Коаксиальный кабель, витая пара, оптоволокно. Радиоволны, микроволны, инфракрасное излучение

При построении сети необходимо, прежде всего, определить, при помощи какого носителя следует передавать связные сигналы, которые принято называть слаботочными.

Под средой передачи данных понимают физическую субстанцию, по которой происходит передача электрических сигналов, использующихся для переноса той или иной информации, представленной в цифровой форме.

Классификация по типу среды передачи:

1) проводные сети, то есть сети, каналы связи которых построены с использованием медных или оптических кабелей;

2) беспроводные сети, то есть сети, в которых для связи используются беспроводные каналы связи, например, радио, СВЧ, инфракрасные или лазерные каналы.

Кабельные среды передачи данных обеспечивают передачу сигнала по строго определенному пути. Наиболее широко используемые в настоящее время кабельные среды передачи данных представлены кабелями следующих типов: коаксиальный кабель, витая пара и оптический кабель.

Коаксиальный кабель. Этот кабель представляет собой медный проводник, по которому передается полезный сигнал. Проводник окружен изоляцией, поверх которой укладывается медная фольга или сетка, представляющая собой экран, защищающий центральный сигнальный провод от внешних электромагнитных помех (рис 2.9). Благодаря использованию такой конструкции экран обеспечивает высокую степень защиты полезного сигнала от внешних помех, что позволяет без существенных потерь осуществлять передачу сигнала на достаточно большие расстояния. Существующие коаксиальные кабели подразделяют на два типа: тонкий и толстый.

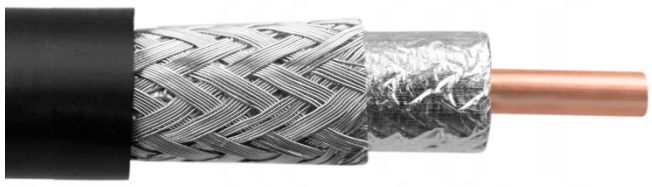


Рисунок 2.9. – Коаксиальный кабель

Тонкий коаксиальный кабель внешне очень похож на современные кабели, используемые для подключения телевизионных антенн. Такой кабель

не настолько гибок и удобен при монтаже, как неэкранированная витая пара, но тоже достаточно часто используется для построения локальных сетей. Разъемы, используемые для подключения тонкого коаксиального кабеля, называются BMC-разъемами.

Толстый коаксиальный кабель очень похож на тонкий, но только он большего диаметра. Увеличение диаметра кабеля позволяет обеспечить его большую помехоустойчивость и соответственно гарантирует возможность передачи полезного сигнала на большие расстояния, чем тонкий коаксиальный кабель. Из-за более сложного процесса монтажа толстого кабеля (плохо гнется и требует специализированных разъемов) он распространен гораздо меньше.

Витая пара. Этот кабель состоит из двух или более медных проводников, защищенных пластиковой изоляцией и свитых между собой (рис. 2.10). Свитые проводники снаружи защищаются еще одним слоем изоляции. Свивание проводников уменьшает искажение полезного сигнала, связанное с передачей электрического тока по проводнику. С точки зрения физики процесс такого искажения называется интерференцией сигналов.

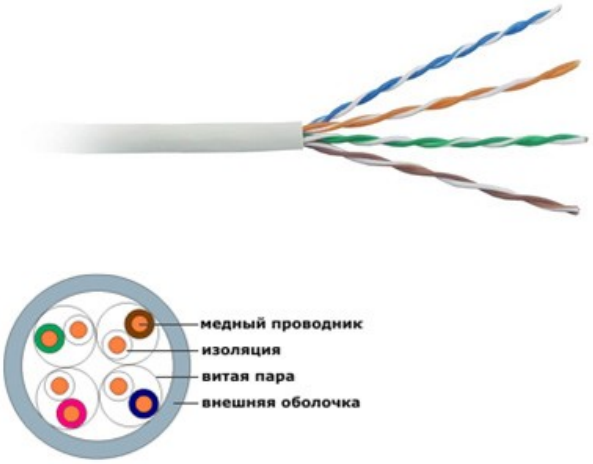


Рисунок 2.10. – Витая пара

В настоящее время существует несколько вариаций кабелей типа «витая пара»: экранированная витая пара и неэкранированная витая пара. При производстве экранированной витой пары свитые между собой проводники снаружи окружаются дополнительной металлической оболочкой - экраном. Эта дополнительная оболочка обеспечивает защиту полезного сигнала, передающегося по витой паре от внешних электромагнитных помех. Неэкранированная витая пара не имеет дополнительного внешнего металлического экрана. Для соединения кабелей на основе неэкранированной витой пары используются разъемы RJ-45 (рис. 2.11). Внешне они очень похожи на разъемы, используемые для подключения телефонного кабеля.

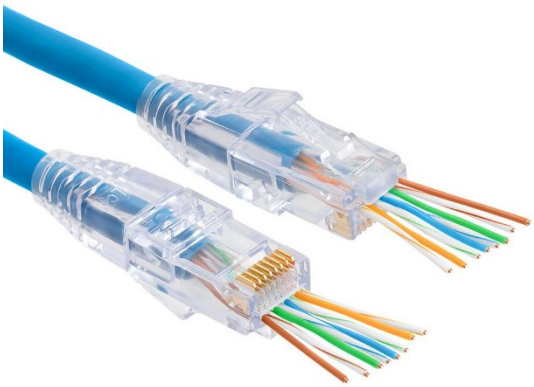


Рисунок 2.11. – Rj45 разъем Cat5e

Витая пара является одной из самых распространенных видов кабеля в сфере телекоммуникаций, остановимся на ее рассмотрении поподробнее.

Ее высокая популярность обусловлена:

- невысокой стоимостью,
- универсальностью,
- высокой стабильностью передачи сигнала.

Можно классифицировать кабель по типу жил:

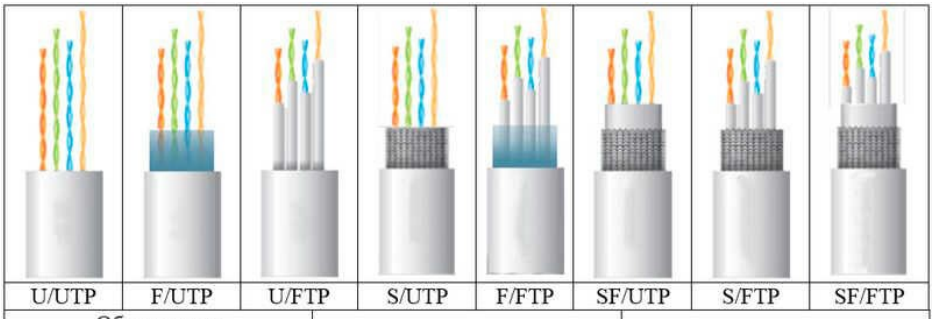
Одножильный – каждый провод состоит всего из одной медной жилы толщиной 0,3-0,6 мм. Главный недостаток – низкая износостойкость. При периодическом изгибании в одном и том же месте медные проводники быстро ломаются.

Многожильный – каждая жила состоит из множества тончайших проволочек и не ломается при сгибании и скручивании, однако из-за небольшого сечения проводников сигнал в них затухает быстрее, поэтому общая длина кабеля не может быть больше 100 м.

Скручивание жил уменьшает наводки, но полностью не устраняет их, поэтому еще одно отличие кабеля витая пара заключается в наличии одного или нескольких металлических экранов. Об этом свидетельствует маркировка (рис 2.12):

- ***UTP*** – самый простой и дешевый тип без дополнительного экранирования. Именно он, в основном, используется интернет-провайдерами при проведении интернета в квартиры.
- ***FTP*** (он же ***F/UTP***) – кабель с общим экраном из фольги.
- ***STP (S/UTP)*** – с общим экраном из металлической оплетки.
- ***S/FTP*** – с двойным экраном из фольги и оплетки.
- ***U/FTP*** – с фольгированием каждой пары, но без общего экрана.
- ***S/FTP*** – с фольгированием каждой пары и общей оплеткой.
- ***F/FTP*** – с фольгированием пар и общим экраном из фольги.

•***SF/FTP*** – с фольгированием пар и двойным общим экраном из фольги и оплетки.



Обозначение по ISO/IEC 11801	Общий экран	Экран для свитых пар
U/UTP	нет	нет
F/UTP	фольга	нет
U/FTP	нет	фольга
S/UTP	оплётка	нет
F/FTP	фольга	фольга
SF/UTP	оплётка, фольга	нет
S/FTP	оплётка	фольга
SF/FTP	оплётка, фольга	фольга

Рисунок 2.12. – Типы витой пары

Экранированные типы, в частности FTP, обычно используются для создания ***патч-кордов (соединительных шнуров)*** в локальных сетях с высокой пропускной способностью или монтаже линий с высокой плотностью кабелей.

Помимо различных видов она также делится на категории с цифро-буквенной маркировкой (рис 2.13). Всего их 10, и чем выше номер, тем выше надежность и пропускная способность.

•***UTP Cat.1*** – две жилы без скрутки и экранирования. Устарела, не используется.

•***UTP Cat.2*** – 4 жилы. Устарела из-за малой полосы пропускания.

•***UTP Cat.3*** – 8 жил, скрученных попарно. Обеспечивает пропускную способность до 100 Мбит/с. Максимальная длина линии – 100 м. Сейчас, в основном, применяется для прокладки телефонных линий.

•***UTP Cat.4*** – 8 жил. Пропускная способность – до 16 Мбит/с. Устарела, не используется.

•***UTP Cat.5*** – 4 скрутки. Максимальная скорость – до 100 Мбит/с при использовании 2 пар и до 1 Гбит/с – при 4.

•***UTP Cat.5E*** – улучшенная «версия» предыдущей категории. Более тонкий и дешевый кабель, но не менее надежный. Самый популярный в настоящий момент.

•***UTP Cat.6E*** – 4 скрутки без экрана. Пропускная способность – до 10 Гбит/с на линии длиной до 55 м. Вторая по популярности категория.

• ***UTP Cat.6A*** – 4 скрутки, S/FTP или F/FTP. Пропускная способность та же, но допустимая длина увеличена до 100 м.

• ***UTP Cat.7*** – аналог прошлой категории с измененной частотной полосой.

• ***UTP Cat.7A*** – 4 скрутки, S/FTP или F/FTP. Максимальная скорость – 40 Гбит/с по линии до 50 м и 100 Гбит/с – по линии до 15 м.

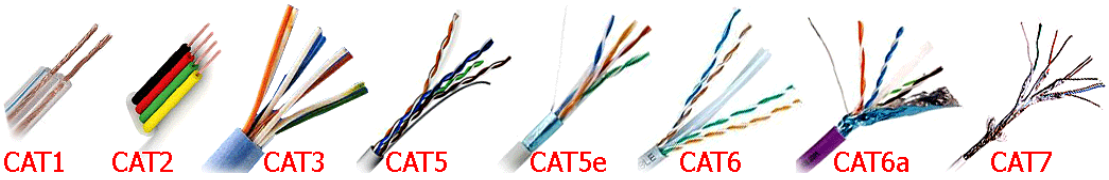


Рисунок 2.13. – категории витой пары

Оптический кабель. Он используется для передачи сигнала в виде световых импульсов. Оптический кабель обеспечивает очень низкие потери полезного сигнала и за счет этого позволяет передавать данные на очень большие расстояния (в настоящее время до нескольких десятков километров). В дополнение к этому благодаря использованию света в качестве сигнала обеспечивается полная защищенность от внешних электромагнитных помех. На рис. 2.14 представлена конструкция оптического кабеля ОК-М.

Рисунок 2.14. – Оптический кабель

В качестве проводника в таких кабелях используется стеклянное или пластиковое волокно, защищенное снаружи изоляцией для обеспечения физической сохранности. Оптическое волокно является относительно дорогой средой передачи (по сравнению с витой парой и коаксиальным кабелем), но в настоящее время активно используется для построения высокоскоростных и протяженных линий связи.

Принцип системы оптической связи заключается в передаче сигнала через оптоволокно к удаленному приёмнику. Электрический сигнал

преобразуется в оптический и в таком виде передаётся на расстояние. В приёмном устройстве он обратно переходит в исходную электрическую форму. У волоконно-оптической связи есть множество преимуществ перед другими типами передачи информации, такими как медные жилы и системы радиосвязи.

• Сигнал может быть передан без регенерации на большое расстояние (200 км).

• Оптоволоконная передача не чувствительна к электромагнитным помехам. Кроме того, волокно не проводит электричество и фактически нечувствительно к радиочастотной интерференции.

• Оптические системы обеспечивают большее количество каналов чем физические цепи.

• Оптический кабель намного легче и тоньше чем кабель с металлическими жилами и волокна занимают в нём небольшой объём. Например, один оптоволоконный кабель может содержать 144 волокна.

• Оптическое волокно очень надёжно.

• У оптического волокна срок эксплуатации больше 25-и лет (по сравнению с 10 годами систем спутниковой связи).

• Рабочие температуры для оптического волокна изменяются, но они обычно они лежат в диапазоне от -40° до +80°C

Группа факторов ухудшают пропускание света в оптической системе связи:

1. ***Затухание***: Поскольку световой сигнал перемещается через волокно, он теряет мощность из-за поглощения, рассеивания, и других потерь. С некоторым расстоянием мощность сигнала может уменьшиться до уровня собственных шумов приёмника.

2. ***Пропускная способность***: Оптоволокно имеет ограниченный частотную полосу пропускания и если световой сигнал использует несколько частот, то это явление уменьшает информационную пропускную способность.

3. ***Дисперсия***: Импульсы света распространяющиеся в волокне расширяются и тем ограничивают информационную пропускную способность на высоких скоростях передачи или укорачивается её расстояние.

Для сетей оптической линии связи используется два типа оптического волокна: одномодовое и многомодовое.

Одномод (SingleModeFiber - SMF) передает только один несущий световой сигнал или одну моду. Из-за малого диаметра сердечника световой сигнал способен проходить через одномодовое оптическое волокно только по одному пути («моде»).

Многомод (MultiModeFiber MMF) передает несколько независимых световых сигналов (мод) с разными длинами волн и фазами. В многомодовом оптическом волокне диаметр сердечника гораздо больше, чем в одномодовом, поэтому один и тот же световой сигнал проходит по нескольким разным путям («модам»).

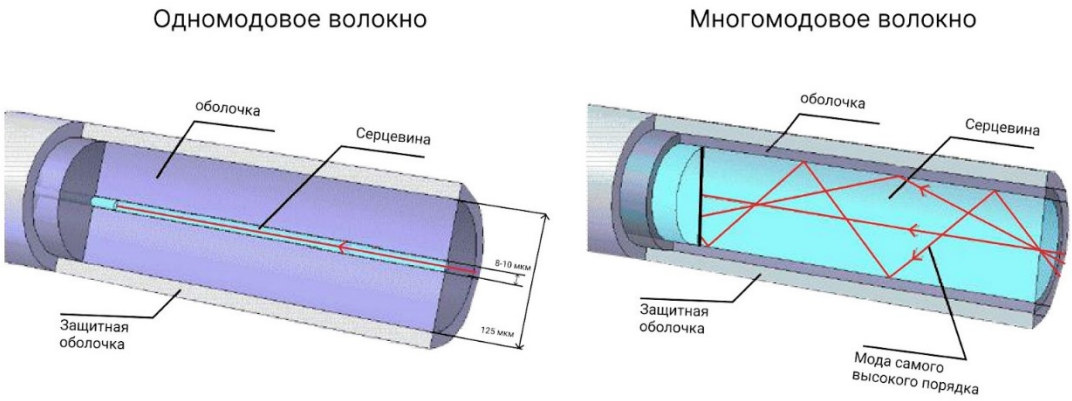


Рисунок 2.15. – Типы оптического волокна

Основное отличие в конструкции одномода и многомода – диаметр сердечника. У одномодового оптоволокна d сердечника не превышает 10 мкм, у многомодового d равен 50 мкм и 62,5 мкм (рис 2.12).

Рассмотрим ряд современных беспроводных технологий, используемых при создании передачи данных и организации сети.

Радиосеть - беспроводная сеть с радиоканалами, в которой передача данных осуществляется с помощью волн, электромагнитный спектр которых охватывает область от нескольких герц до сотен и тысяч Гц.

Передача происходит следующим образом: на передающей стороне (в ***радиопередатчике*** - устройство для формирования радиочастотного сигнала, подлежащего излучению) формируются высокочастотные колебания (***несущий сигнал*** - сигнал, один или несколько параметров которого подлежат изменению в процессе модуляции.) определенной частоты. На него накладывается сигнал, который нужно передать (звуки, изображения и т. д.) — происходит ***модуляция*** (процесс изменения одного или нескольких параметров высокочастотного несущего колебания по закону низкочастотного информационного сигнала) несущей полезным сигналом. Сформированный таким образом высокочастотный сигнал излучается антенной в пространство в виде радиоволн. На приёмной стороне радиоволны наводят модулированный сигнал в приемной антенне, он поступает в радиоприёмник. Здесь система фильтров выделяет из множества наведенных в антенне токов от разных передатчиков сигнал с нужной несущей частотой, а ***детектор*** (электронный узел устройств, отделяющий полезный (модулирующий) сигнал от несущей

составляющей.) выделяет из него модулирующий полезный сигнал. Получаемый сигнал может несколько отличаться от передаваемого передатчиком вследствие влияния разнообразных помех.

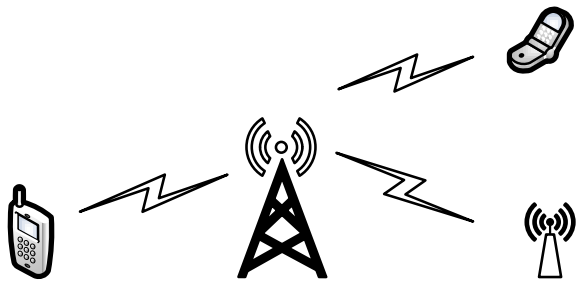


Рисунок 2.16. – Радиосеть

Радиорелейные линии (сеть) (РРЛ) представляют собой цепочку приемопередающих радиостанций (оконечных, промежуточных, узловых), которые осуществляют последовательную многократную ретрансляцию (прием, преобразование, усиление и передачу) передаваемых сигналов.

В зависимости от используемого вида распространения радиоволн РРЛ можно разделить на две группы: прямой видимости и тропосферные.

РРЛ прямой видимости являются одним из основных наземных средств передачи сигналов телефонной связи, программ звукового и ТВ вещания, цифровых данных и других сообщений на большие расстояния. Ширина полосы частот сигналов многоканальной телефонии и ТВ составляет несколько десятков мегагерц, поэтому для их передачи практически могут быть использованы диапазоны только дециметровых и сантиметровых волн, ширина спектра которых составляет 30ГГц. Кроме того, в этих диапазонах почти полностью отсутствуют атмосферные и промышленные помехи. Расстояние между соседними станциями (протяженность пролета) R зависит от рельефа местности и высоты подъема антенн.

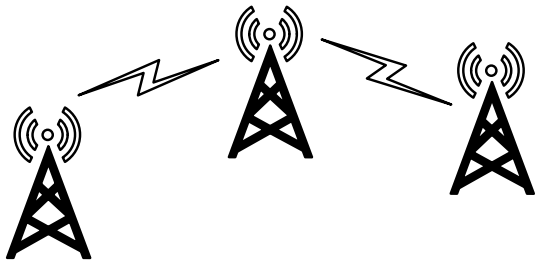


Рисунок 2.17. – Радиорелейная связь

Спутниковая связь — один из видов космической радиосвязи,

основанный на использовании в качестве ***ретрансляторов*** (оборудование связи, которое соединяет два или более радиопередатчика, удалённых друг от друга на большие расстояния) искусственных спутников Земли, как правило специализированных спутников связи. Спутниковая связь осуществляется между так называемыми земными станциями, которые могут быть как стационарными, так и подвижны.

Принцип спутниковой космической связи предполагает передачу/прием радиосигнала с использованием базовых наземных или подвижных станций через спутниковый ретранслятор (рис 2.15). Данная специфика обеспечения прохождения радиоволн обусловлена кривизной земной поверхности, препятствующей прохождению радиосигнала. Иными словами, в зоне прямой видимости радиосигнал с одной станции на другую транслируется без задержек. Однако, если стоит задача получить сигнал за многие тысячи километров от передающей станции, то требуется ретранслятор, направляющий сигнал под соответствующим углом на приемную станцию.

По своей сути, спутниковая связь через устройство-ретранслятор является типовой аналогией радиорелейной связи, только в этом случае, ретранслятор располагается на значительном расстоянии (высоте) от земной поверхности, исчисляемой тысячами километров. Если для организации радиосвязи на большие расстояния в разные места земного шара требовалось множество наземных ретрансляторов, то с появлением космических спутников их количество сократилось в разы. Теперь для трансляции радиосигнала с одной материковой части на другую требуется всего один спутник.

Спутниковая связь, в целом, обеспечивается целым комплексом взаимосвязанных элементов системы связи: спутниками-ретрансляторами; стационарными земными станциями спутниковой связи на земной поверхности; центром управления спутниковой связи (ЦУСС) и др. элементами системы.

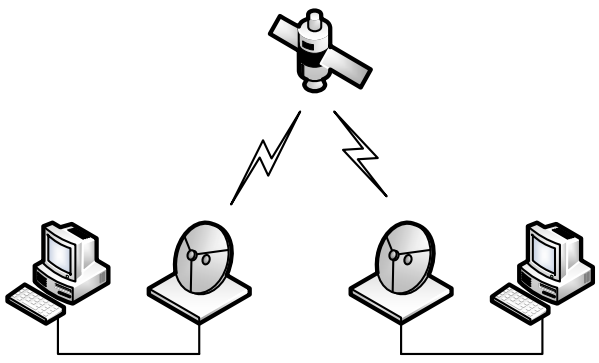


Рисунок 2.18. – Спутниковая связь

Транкинговые системы радиосвязи - это системы связи, в которых выполняется автоматическое и динамическое распределение каналов между пользователями. Современные транкинговые системы связи обеспечивают разные типы вызовов: групповые, индивидуальные, приоритетные. Транковая связь широко используется корпоративными и частными клиентами в различных отраслях.

Инфраструктура транкинговой системы представлена базовой станцией (БС), в состав которой, помимо радиочастотного оборудования (ретрансляторы, устройство объединения радиосигналов, антенны), входят также коммутатор, устройство управления и интерфейсы различных внешних сетей. Необязательными, но очень характерными элементами инфраструктуры транкинговой системы связи являются диспетчерские пульты. Транкинговые системы используются в первую очередь теми потребителями, чья работа не обходится без диспетчера (рис 2.19).

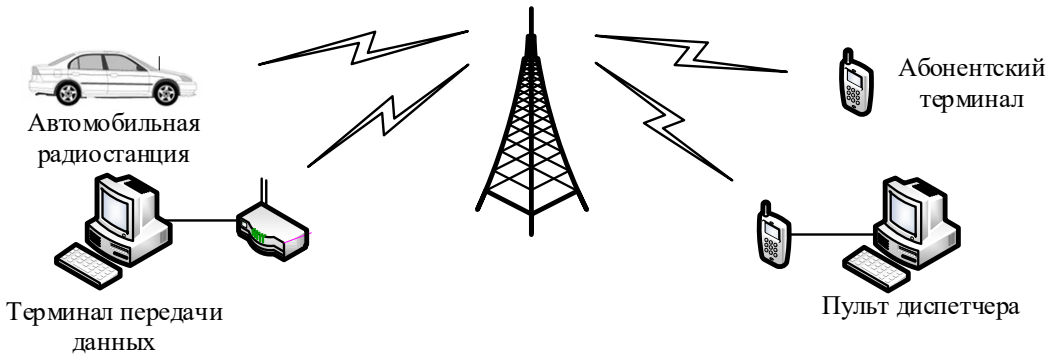


Рисунок 2.19. – Транкинговая связь

На текущий момент, выпускается большое количество систем транкинговой связи подходящих под самые разнообразные задачи заказчиков. Всех их можно разделить по следующим параметрам:

По способу передачи голосовых сообщений:

- аналоговые (Smartrunk II, Smartlink, EDACS, LTR, MPT 1327);
- цифровые (EDACS, APCO 25, TETRA, Tetrapol).

По организации доступа к системе:

- без канала управления (Smartrunk II);
- с распределенным каналом управления (LTR, Smartlink);
- с выделенным каналом управления (MPT 1327).

По способу удержания канала:

- с удержанием канала на весь сеанс переговоров (Smartrunk II, MPT);
- с удержанием канала на время одной передачи (LTR, Smartlink).

По конфигурации радиосети:

- однозоновые системы (Smartrunk)

- многозоновые системы (MPT, LTR, Smartlink, TETRA, APCO, EDACS, tetrapol)

По способу организации радиоканала:

- полудуплексные (Smartrunk II, MPT 1327, LTR, Smartlink, TETRA, APCO25, TETRAPOL);
- дуплексные (TETRA, APCO25, TETRAPOL).

Транкинговая связь обеспечивает высокую оперативность установления соединения (0,2-1 сек), доступ к системе исходя из установленных приоритетов и экстренное предоставление канала связи абоненту с более высоким приоритетом; меньшие затраты на развертывание и эксплуатацию систем, экономия частотных ресурсов, более высокий уровень сервиса - индивидуальные вызовы, приоритеты, интеграция с другими сетями; возможность передачи цифровых данных; покрытие связью больших площадей благодаря многозоновой конфигурации.

Инфракрасная беспроводная сеть — не требует для своего функционирования проводных соединений. В компьютерной технике обычно используется для связи компьютеров с периферийными устройствами (интерфейс IrDA)

В отличие от радиоканала, инфракрасный канал нечувствителен к электромагнитным помехам, и это позволяет использовать его в производственных условиях. У инфракрасного канала высокая стоимость приемников и передатчиков, где требуется преобразование электрического сигнала в инфракрасный и обратно, а также низкие скорости передачи (обычно не превышает 5-10 Мбит/с, но при использовании инфракрасных лазеров возможны существенно более высокие скорости). В условиях прямой видимости инфракрасный канал может обеспечить связь на расстояниях в несколько километров, но наиболее удобен он для связи компьютеров, находящихся в одном помещении, где отражения от стен комнаты дает устойчивую и надежную связь. Наиболее естественный тип топологии здесь — «шина» (то есть переданный сигнал одновременно получают все абоненты). Имея такое количество недостатков, инфракрасный канал не смог получить широкого распространения.

Все инфракрасные беспроводные сети используют для передачи данных инфракрасные лучи. В подобных системах необходимо генерировать очень сильный сигнал, т. к. на него оказывают влияние другие источники.

Инфракрасные сети нормально функционируют на скорости 10 Мбит/с. Различают 4 типа инфракрасных сетей:

- Сети прямой видимости (между приемником и передатчиком).
- Сети на рассеянном излучении. Сигнал отражается от стен и потолка и,

в конце концов, достигает приемника. Дальность до 30 м и не высокая скорость передачи данных.

•Сети на отраженном излучении. Оптические трансиверы компьютеров передают сигналы в определенное место, откуда они переадресуются другому компьютеру.

•Широкополосные оптические сети предоставляют услуги, соответствующие жестким требованиям мультимедийной среды и практически не уступают кабельным системам.

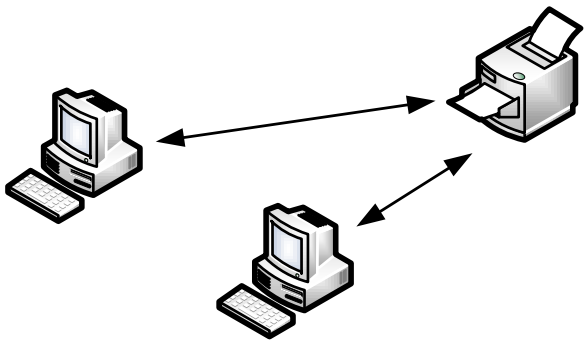


Рисунок 2.20. – Инфракрасная сеть

Сотовая сеть или мобильная сеть - это сеть связи, в которой связь с конечными узлами и от них является беспроводной. Сеть распределена по наземным участкам, называемым "ячейками", каждая из которых обслуживается по меньшей мере одним приемопередатчиком фиксированного местоположения (обычно тремя сотовыми узлами или базовыми приемопередающими станциями).

Сеть составляют разнесённые в пространстве приёмопередатчики, работающие в одном и том же частотном диапазоне, и коммутирующее оборудование, позволяющее определять текущее местоположение подвижных абонентов и обеспечивать непрерывность связи при перемещении абонента из зоны действия одного приёмопередатчика в зону действия другого.

Основные составляющие сотовой сети — это сотовые телефоны и базовые станции, которые обычно располагают на крышах зданий и вышках. Будучи включённым, сотовый телефон прослушивает эфир, находя сигнал базовой станции (рис. 2.21). После этого телефон посылает станции свой уникальный идентификационный код. Телефон и станция поддерживают постоянный радиоконтакт, периодически обмениваясь пакетами. Связь телефона со станцией может идти по аналоговому протоколу (AMPS, NAMPS, NMT-450) или по цифровому (DAMPS, CDMA, GSM, UMTS). Если телефон выходит из поля действия базовой станции (или качество радиосигнала

сервисной соты ухудшается), он налаживает связь с другой (англ. handover).

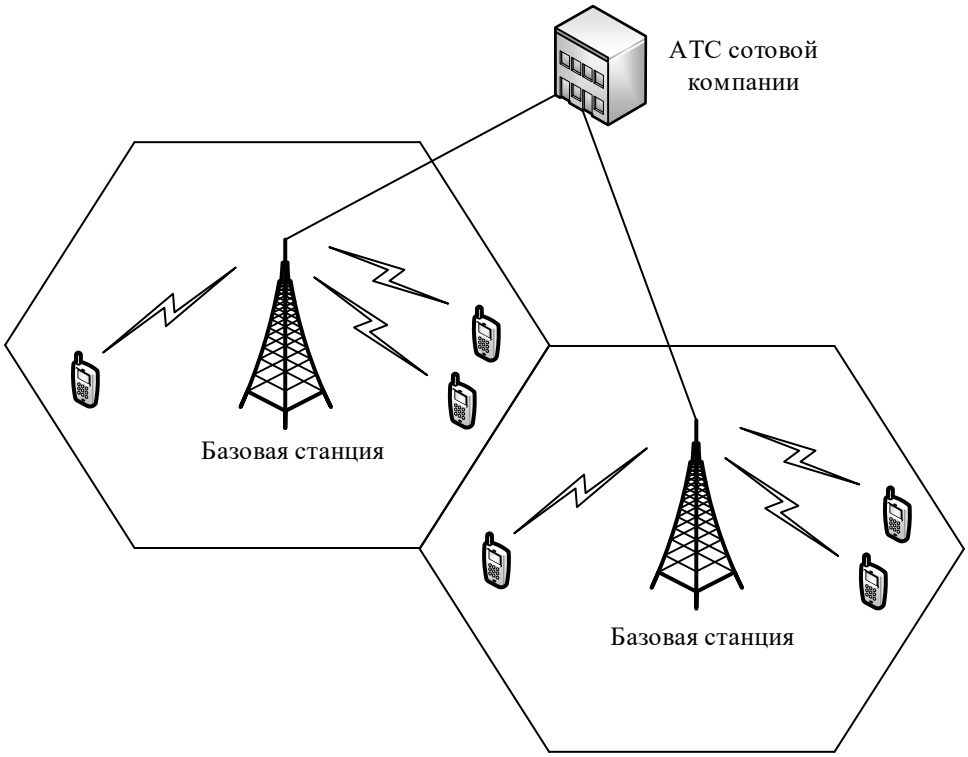


Рисунок 2.21. – Сотовая сеть

Сотовые сети могут состоять из базовых станций разного стандарта, что позволяет оптимизировать работу сети и улучшить её покрытие.

Сотовые сети разных операторов соединены друг с другом, а также со стационарной телефонной сетью. Это позволяет абонентам одного оператора делать звонки абонентам другого оператора, с мобильных телефонов на стационарные и со стационарных на мобильные.

Операторы могут заключать между собой договоры роуминга. Благодаря таким договорам абонент, находясь вне зоны покрытия своей сети, может совершать и принимать звонки через сеть другого оператора. Как правило, это осуществляется по повышенным тарифам. Возможность роуминга появилась лишь в стандартах 2G и является одним из главных отличий от сетей 1G.

Операторы могут совместно использовать инфраструктуру сети, сокращая затраты на развертывание сети и текущие издержки.

2.4. Методы доступа к среде передачи: конфликтные и бесконфликтные

Метод доступа – это способ определения того, какая из рабочих станций сможет следующей использовать ЛВС. То, как сеть управляет доступом к

каналу связи (кабелю), существенно влияет на ее характеристики.

Методы доступа к среде передачи бывают ***конфликтные*** (состязательные) и ***неконфликтные***. Конфликтные методы доступа предполагают возможность коллизии (collision – столкновение), т.е. одновременной передачи по одной линии двумя и более компьютерами.

Конфликтные методы доступа:

- CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance)
- CSMA/CD (Carrier Sense Multiple Access/Collision Detect)

Неконфликтные методы доступа:

- метод доступа с маркером
- метод доступа по приоритету

Метод доступа к передающей среде CSMA/CD (Carrier Sense Multiple Access with Collision Detection), или метод множественного доступа с контролем несущей частоты и обнаружения коллизий, является улучшенной модификацией протокола CSMA. Этот метод используется во всех существующих сетях Ethernet, Fast Ethernet и Gigabit Ethernet, работа которых описана спецификацией IEEE 802.3.

При использовании метода CSMA часто происходит так, что, прослушав канал на наличие синхронизирующего сигнала и не обнаружив такового (то есть линия считается «чистой»), передачу данных могут произвести сразу несколько компьютеров, что, естественно, вызовет коллизии, и данные будут потеряны.

Согласно методу CSMA/CD, прослушивание линии происходит постоянно, при этом если передаваемые сигналы и наблюдаемые сигналы не совпадают, значит, кто-то еще делает попытку передачи данных. В этом случае, чтобы избежать коллизий и потери данных, передача данных временно прекращается, и отправитель отправляет в линию специальный сигнал jam (32-битная последовательность), который информирует все остальные компьютеры о том, что уже ведется передача данных и компьютерам запрещено осуществлять аналогичные действия. По истечении случайного промежутка времени происходит повторная попытка передачи данных. С каждой новой попыткой время ожидания увеличивается, но если после 16 последовательных попыток передача данных не будет возобновлена, фиксируется ошибка, говорящая о том, что канал передачи данных недоступен, и сообщение об этом поступает протоколу верхнего уровня.

Благодаря такому подходу передача данных происходит не только быстрее (не нужно повторно передавать весь объем), но и с большей гарантией того, что они будут доставлены, даже несмотря на то, что за качество доставки отвечает уровень LLC.

Метод доступа к передающей среде CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance), или метод множественного доступа с контролем несущей частоты и избеганием коллизий, также является модификацией протокола CSMA. Данный метод доступа к среде чаще всего используется в беспроводных сетях, работа которых описана спецификацией IEEE 802.11.

В отличие от метода доступа к среде CSMA/CD, в которой jam-сигнал высылается только при обнаружении коллизии, метод CSMA/CA сначала отправляет jam-сигнал, информирующий о том, что станция хочет передавать данные, и только потом передает сигнал. После того как выслан jam-сигнал, станция еще некоторое время ожидает и проверяет канал на наличие аналогичных jam-пакетов. Если таковой обнаружен, то есть кто-то уже ведет вещание, станция ждет случайный промежуток времени, и затем процесс повторяется. Если никаких чужих передач не обнаружено, станция начинает передавать данные до тех пор, пока все они не будут переданы. При таком подходе, даже если будет обнаружен чужой jam-пакет, это приведет не к коллизии при передаче данных, а лишь к коллизии jam-пакетов.

Достоинства конфликтных методов доступа:

- простота реализации;
- все сетевые узлы передают и принимают сигналы в одной полосе частот;
- имеется один канал для передачи всех данных;

Недостатки конфликтных методов доступа:

- только один узел может передавать данные в отдельный момент времени;
- узел может либо передавать либо принимать данные (работать в полудуплексном режиме).

Метод доступа по приоритету. В случае метода доступа по приоритету кадры передаются не всем узлам сети, а только узлу назначения. В сети есть выделенный арбитр доступа – концентратор. Концентраторы циклически выполняют опрос портов. Узел, желающий передать пакет, посылает сигнал концентратору, запрашивая передачу кадра и указывая его приоритет. Есть два уровня приоритетов: высокий и низкий. Высокий соответствует данным, чувствительным к временным задержкам (речь, видео). Если сеть свободна, концентратор разрешает передачу пакета.

Метод доступа с маркером. В среде постоянно циркулирует специальное служебное сообщение, называемое маркером (token).

Право на передачу имеет только тот узел, который в данный момент владеет маркером. Узел захвативший маркер загружает в него пакет и отправляет его в путь снабдив его адресами.

Недостатки данного метода:
- более медленный метод;
- плохо приспособлен к неравномерной загрузке сети.

Методы доступа с маркером:
- Token Ring (IBM);
- ArcNet (Datapoint);
- FDDI.

2.5. Модель взаимодействия открытых систем

Обмен информацией между компьютерами, объединенными в сеть, очень сложная задача. Это связано с тем, что существует много производителей аппаратных и программных средств вычислительных систем. Единственный выход — унифицировать средства сопряжения систем, а именно использовать открытые системы. Открытая система взаимодействует с другими системами на основе единых общедоступных стандартов и спецификаций.

Международная Организация по Стандартизации (ISO) представила индустриальный стандарт — модель взаимодействия открытых систем (Open System Interconnection Reference Model — OSI/RM, чтобы помочь поставщикам создавать совместимые сетевые аппаратные и программные средства. В соответствии с этой моделью выделяются следующие уровни (рис.1.4)

В соответствии с эталонной моделью OSI задача обмена информацией между компьютерами в сети разбивается на ряд относительно независимых и менее сложных подзадач взаимодействия между смежными уровнями (рис. 2.22).

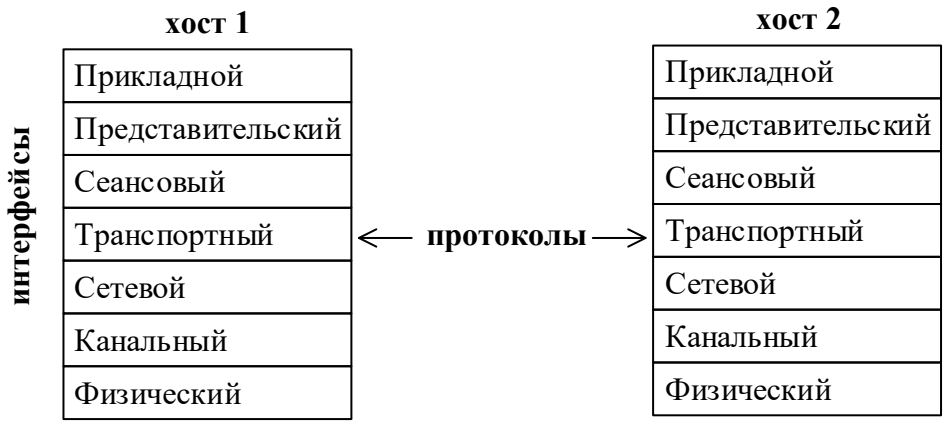


Рисунок 2.22. – Взаимодействие между уровнями OSI

Связь между уровнями двух сетевых узлов (горизонтальное взаимодействие) выполняется в соответствии с унифицированными правилами — протоколами взаимодействия.

В автономной системе передача данных между уровнями (вертикальное взимодействие) реализуется через интерфейсы API.

Границу между сеансовым и транспортным уровнями можно рассматривать как границу между протоколами прикладного уровня и протоколами низших уровней. Если прикладной, представительный и сеансовый уровни обеспечивают прикладные процессы сеанса взаимодействия, то четыре низших уровня решают проблемы транспортировки данных.

Два самых низших уровня — физический и канальный — реализуются аппаратными и программными средствами, остальные пять более высоких уровней реализуются, как правило, программными средствами.

При передаче информации от прикладного процесса в сеть на физический уровень происходит ее обработка, которая заключается в разбиении передаваемых данных на отдельные блоки, преобразовании формы представления или кодировки данных в блоке и добавлении к каждому блоку заголовка (header) соответствующего уровня. Каждый заголовок характеризует используемый протокол обработки данных, причем каждый уровень воспринимает в качестве данных весь блок, полученный от предыдущего уровня, включая присоединенный заголовок. Такое построение эталонной модели позволяет заложить (инкапсулировать) в каждый передаваемый по физической среде информационный блок сведения, необходимые для выбора последовательности протоколов для осуществления обратных преобразований на принимающей информацию стороне.

Физический уровень определяет такие характеристики соединения, как уровни напряжений, синхронизацию и физическую скорость передачи данных, максимальные расстояния передачи, конструктивные параметры разъемов и другие аналогичные характеристики. Известные стандарты RS-232-C, V.24 и IEEE 802.3 (Ethernet).

Канальный уровень отвечает за надежную передачу данных через физический канал, а именно:

- обеспечивает физическую адресацию (в отличие от сетевой или логической адресации);
- обеспечивает обнаружение ошибок в передаче и восстановление данных;
- отслеживает топологию сети и обеспечивает дисциплину использования сетевого канала конечной системой;

•обеспечивает уведомление о неисправностях;

•обеспечивает упорядоченную доставку блоков данных и управление потоком информации.

Для ЛВС канальный уровень разбивается на два подуровня:

LLC (***Logical Link Control***) — обеспечивает управление логическим звеном, т.е. собственно функции канального уровня;

MAC (***Media Access Control***) — обеспечивает специальные методы доступа к среде распространения.

Сетевой уровень протоколы маршрутизации сети которого позволяют выбирать оптимальные маршруты через связанные между собой подсети.

Транспортный уровень обеспечивает высшим уровням услуги по транспортировке данных, а именно:

•обеспечивает надежную транспортировку данных через объединенную сеть;

•обеспечивает механизмы для установки, поддержания и упорядоченного завершения действия виртуальных каналов;

•обеспечивает обнаружение и устранение неисправностей транспортировки;

•следит за тем, чтобы конечная система не была перегружена слишком большим количеством данных.

Сеансовый уровень синхронизирует диалог между объектами представительного уровня, определяет точки синхронизации для промежуточного контроля и восстановления при передаче файлов. Этот уровень также позволяет производить обмен данными в режиме, заданном прикладной программой, или предоставляет возможность выбора режима обмена.

Кроме основной функции управления диалогом сеансовый уровень предоставляет средства для выбора класса услуг и уведомления об исключительных ситуациях (проблемах сеансового, представительного и прикладного уровней).

Представительный уровень обеспечивает служебные операции, выбираемые на прикладном уровне, для интерпретации передаваемых и получаемых данных: управление информационным обменом, отображение данных и управление структурированными данными. Эти служебные данные позволяют связывать воедино терминалы и вычислительные средства различных типов. Примером протокола этого уровня является XDR.

Прикладной уровень — самый близкий к пользователю уровень OSI — не предоставляет услуги другим уровням OSI, однако он обеспечивает прикладные процессы, лежащие за пределами масштаба модели OSI, а так же:

•идентифицирует и устанавливает наличие предполагаемых партнеров для связи;

•синхронизирует совместно работающие прикладные программы;

•устанавливает соглашение по процедурам устранения ошибок и управления целостностью информации;

•определяет достаточность наличных ресурсов для предполагаемой связи.

Модель OSI не является реализацией, она лишь предлагает порядок организации взаимодействия между компонентами системы. Реализациями этих правил являются стеки протоколов.

Протоколы стека OSI и их распределение по уровням сетевой модели приведены рисунке 2.23

<table>
<tr><td>FTAM
ISO 8571</td><td colspan="3">CMISE
ISO 9596
ISO 9595</td><td rowspan="2">Прикладной</td></tr>
<tr><td>ACSE
X.227, ISO 8650
X.217, ISO 8649</td><td>ACSE
X.227, ISO 8650
X.217, ISO 8649</td><td colspan="2">ROSE
ISO 9072
X.219, X.229</td></tr>
<tr><td colspan="4">X.226, ISO 8823
X.209, ISO 8825 BER
X.216, ISO 8822</td><td>Представительский</td></tr>
<tr><td colspan="4">X.225, ISO 8327
X.215, ISO 8326</td><td>Сеансовый</td></tr>
<tr><td colspan="4">X.224, ISO 8073/AD 2
(X.214, ISO 8072/AD 2)
class4</td><td>Транспортный</td></tr>
<tr><td colspan="4">ISO 9542 (ES-IS)
ISO 10589 (IS-IS LEVEL 1)/ISO 10747 (IS-IS LEVEL 2)</td><td rowspan="2">Сетевой</td></tr>
<tr><td colspan="2">ISO 8473-3 (CLNS)
ISO 8208 X.25 Packet Level</td><td>ISO 8473-2
(CLNS)</td><td>ISO 8473-4
(CLNS)</td></tr>
<tr><td colspan="2">ISO 7776 LapB
X.25 Data Link Layer</td><td colspan="2">ISO 802.2 LLC
ISO 802.3 MAC</td><td>Q.921
LapD</td><td>Канальный</td></tr>
<tr><td colspan="2">X.21/X21bis
V.35/G.703
2M TSn</td><td>ISO 802.2
Ethernet</td><td>SDH-DCC
2M TSn</td><td>Физический</td></tr>
</table>

Рисунок 2.23. - Протоколы стека OSI ISO

Стек NetBIOS/SMB. Фирмы Microsoft и IBM совместно работали над сетевыми средствами для персональных компьютеров, поэтому стек протоколов NetBIOS/SMB является их совместным детищем. Средства NetBIOS появились в 1984 году как сетевое расширение стандартных функций базовой системы ввода/вывода (BIOS) IBM PC для сетевой программы PC Network фирмы IBM, которая на прикладном уровне (рис. 2.24) использовала для реализации сетевых сервисов протокол SMB.

<table>
<tr><td rowspan="2">SMB</td><td>Прикладной</td></tr>
<tr><td>Представительский</td></tr>
<tr><td rowspan="2">NetBIOS</td><td>Сеансовый</td></tr>
<tr><td>Транспортный</td></tr>
<tr><td rowspan="3"></td><td>Сетевой</td></tr>
<tr><td>Канальный</td></tr>
<tr><td>Физический</td></tr>
</table>

Рисунок 2.24. - Стек NetBIOS/SMB

Протокол NetBIOS работает на трех уровнях модели взаимодействия открытых систем: сетевом, транспортном и сеансовом. NetBIOS может обеспечить сервис более высокого уровня, чем протоколы IPX и SPX, однако не обладает способностью к маршрутизации. Таким образом, NetBIOS не является сетевым протоколом в строгом смысле этого слова. NetBIOS содержит много полезных сетевых функций, которые можно отнести к сетевому, транспортному и сеансовому уровням, однако с его помощью невозможна маршрутизация пакетов, так как в протоколе обмена кадрами NetBIOS не вводится такое понятие как сеть. Это ограничивает применение протокола NetBIOS локальными сетями, не разделенными на подсети. NetBIOS поддерживает как дейтаграммный обмен, так и обмен с установлением соединений.

Протокол SMB, соответствующий прикладному и представительному уровням модели OSI, регламентирует взаимодействие рабочей станции с сервером. В функции SMB входят следующие операции:

- ***Управление сессиями***. Создание и разрыв логического канала между рабочей станцией и сетевыми ресурсами файлового сервера.

- ***Файловый доступ***. Рабочая станция может обратиться к файл-серверу с запросами на создание и удаление каталогов, создание, открытие и закрытие файлов, чтение и запись в файлы, переименование и удаление файлов, поиск файлов, получение и установку файловых атрибутов, блокирование записей.

- ***Сервис печати***. Рабочая станция может ставить файлы в очередь для печати на сервере и получать информацию об очереди печати.

- ***Сервис сообщений***. SMB поддерживает простую передачу сообщений со следующими функциями: послать простое сообщение; послать широковещательное сообщение; послать начало блока сообщений; послать текст блока сообщений; послать конец блока сообщений; переслать имя

пользователя; отменить пересылку; получить имя машины.

Из-за большого количества приложений, которые используют функции API, предоставляемые NetBIOS, во многих сетевых ОС эти функции реализованы в виде интерфейса к своим транспортным протоколам. В NetWare имеется программа, которая эмулирует функции NetBIOS на основе протокола IPX, существуют программные эмуляторы NetBIOS для Windows и стека TCP/IP.

Стек TCP/IP, называемый также стеком DoD и стеком Интернет, является одним из наиболее популярных стеков коммуникационных протоколов. Стек был разработан по инициативе Министерства обороны США (Department of Defence, DoD) для связи экспериментальной сети ARPAnet с другими сателлитными сетями как набор общих протоколов для разнородной вычислительной среды. Сеть ARPA поддерживала разработчиков и исследователей в военных областях. В сети ARPA связь между двумя компьютерами осуществлялась с использованием протокола Internet Protocol (IP), который и по сей день является основным в стеке TCP/IP и фигурирует в названии стека.

Так как стек TCP/IP был разработан до появления модели взаимодействия открытых систем ISO/OSI, то, хотя он также имеет многоуровневую структуру, соответствие уровней стека TCP/IP уровням модели OSI достаточно условно.

Структура протоколов TCP/IP приведена на рис. 2.25. Протоколы TCP/IP делятся на 4 уровня.

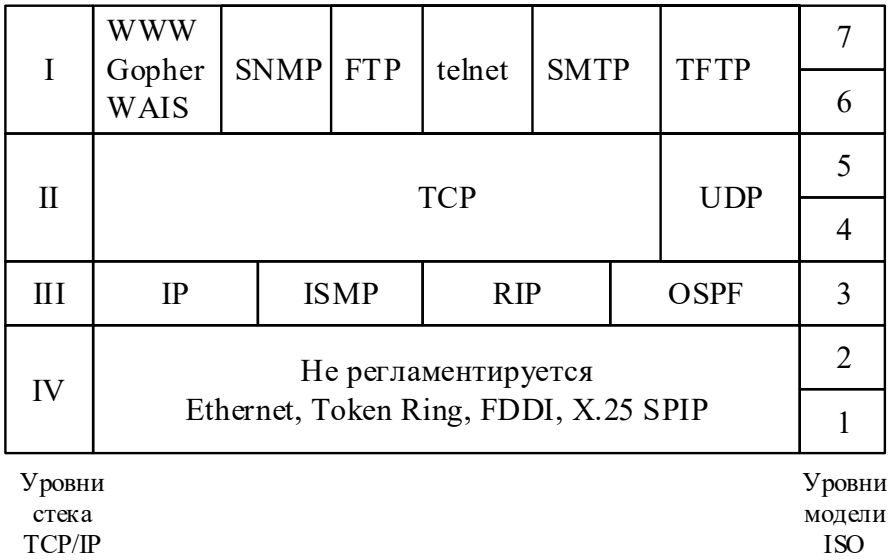


Рисунок 2.25. - Стек TCP/IP

Самый нижний (***уровень IV***) — уровень межсетевых интерфейсов —

соответствует физическому и канальному уровням модели OSI. Этот уровень в протоколах TCP/IP не регламентируется, но поддерживает все популярные стандарты физического и канального уровня: для локальных каналов это Ethernet, Token Ring, FDDI, для глобальных каналов — собственные протоколы работы на аналоговых коммутируемых и выделенных линиях SLIP/PPP, которые устанавливают соединения типа "точка — точка" через последовательные каналы глобальных сетей, и протоколы территориальных сетей X.25 и ISDN. Разработана также специальная спецификация, определяющая использование технологии ATM в качестве транспорта канального уровня.

Следующий уровень (***уровень III***) — это уровень межсетевого взаимодействия, который занимается передачей дейтаграмм с использованием различных локальных сетей, территориальных сетей X.25, линий специальной связи и т. п. В качестве основного протокола сетевого уровня (в терминах модели OSI) в стеке используется протокол IP, который изначально проектировался как протокол передачи пакетов в составных сетях, состоящих из большого количества локальных сетей, объединенных как локальными, так и глобальными связями. Поэтому протокол IP хорошо работает в сетях со сложной топологией, рационально используя наличие в них подсистем и экономно расходуя пропускную способность низкоскоростных линий связи. Протокол IP является дейтаграммным протоколом.

К уровню межсетевого взаимодействия относятся и все протоколы, связанные с составлением и модификацией таблиц маршрутизации, такие как протоколы сбора маршрутной информации ***RIP (Routing Internet Protocol)*** и ***OSPF (Open Shortest Path First)***, а также протокол межсетевых управляющих сообщений ***ICMP (Internet Control Message Protocol)***. Последний протокол предназначен для обмена информацией об ошибках между ***маршрутизатором*** (специализированное устройство, которое пересылает пакеты между различными сегментами сети на основе правил и таблиц маршрутизации) и ***шлюзом*** (аппаратный маршрутизатор или программное обеспечение для сопряжения компьютерных сетей), системой-источником и системой-приемником, то есть для организации обратной связи. С помощью специальных пакетов ICMP сообщается о невозможности доставки пакета, о превышении времени жизни или продолжительности сборки пакета из фрагментов, об аномальных величинах параметров, об изменении маршрута пересылки и типа обслуживания, о состоянии системы и т.п.

Следующий уровень (***уровень II***) называется основным. На этом уровне функционируют протокол управления передачей ***TCP (Transmission Control Protocol)*** и протокол дейтаграмм пользователя ***UDP (User Datagram Protocol)***.

Протокол TCP обеспечивает устойчивое виртуальное соединение между удаленными прикладными процессами. Протокол UDP обеспечивает передачу прикладных пакетов дейтаграммным методом, то есть без установления виртуального соединения, и поэтому требует меньших накладных расходов, чем TCP.

Верхний уровень (***уровень 1***) называется прикладным. За долгие годы использования в сетях различных стран и организаций стек TCP/IP накопил большое количество протоколов и сервисов прикладного уровня: протокол копирования файлов ***FTP***, протоколы удаленного управления ***telnet*** и ***ssh***, почтовый протокол ***SMTP***, гипертекстовые сервисы доступа к удаленной информации, такие как ***WWW*** и многие другие.

2.6. Базовые технологии локальных сетей

Основным разработчиком стандартов локальных сетей является комитет 802, организованный в 1980 году в IEEE. В рамках этого комитета были образованы подкомитеты 802.1, 802.2,…, в которых разрабатываются стандарты разных уровней IEEE-модели и различных технологий построения ЛВС. На рис.2.26 перечислены некоторые из этих стандартов, представляющие собой рекомендации по разработке ЛВС, обеспечивающие выполнение основных требований к организации сетей, таких как открытость, гибкость и совместимость. Стандарты ЛВС обрастают дополнениями, которые находят отражение в обозначениях 802.x в виде букв, например 802.1p (стандарт, описывающий приоритезацию трафика на канальном уровне), а также пополняются новыми стандартами, отражающими появление новых технологий локальных сетей, например беспроводных сетей 802.11 и 802.16.

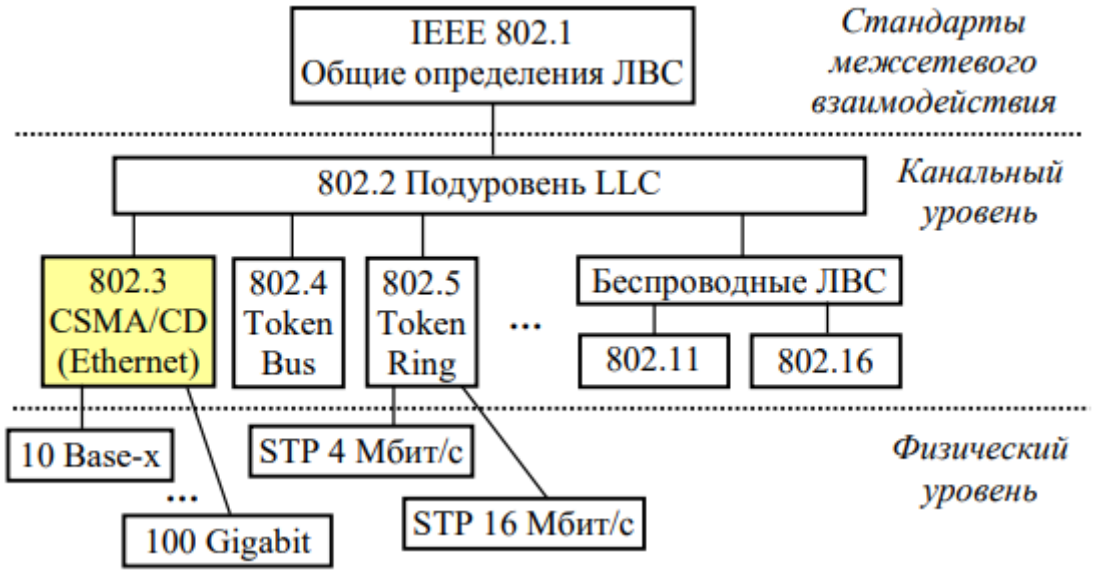


Рисунок 2.26. - Стандарты локальных сетей

Отметим, что канальный уровень находится между сетевым и физическим уровнями модели OSI, поэтому он должен предоставлять сервис вышележащему уровню, взаимодействуя с сетевым протоколом, и обеспечивая инкапсулированным в кадр пакетам доступ к сетевой среде. В то же время, канальный уровень управляет процессом размещения передаваемых данных в физической среде. Поэтому канальный уровень разделен на 2 подуровня (рис. 2.26): верхний подуровень управления логическим каналом передачи данных (***Logical Link Control - LLC***), являющийся общим для всех технологий, и нижний подуровень управления доступом к среде (***Media Access Control - MAC***). Кроме того, средства канального уровня позволяют обнаруживать ошибки в передаваемых данных.

Взаимодействие узлов локальных сетей происходит на основе протоколов канального уровня. Передача данных в локальных сетях происходит на сравнительно короткие расстояния (внутри зданий или между близко расположенными зданиями), но с высокой скоростью (10 Мбит/с - 100 Гбит/с). Расстояние и скорость передачи данных определяется аппаратурой соответствующих стандартов.

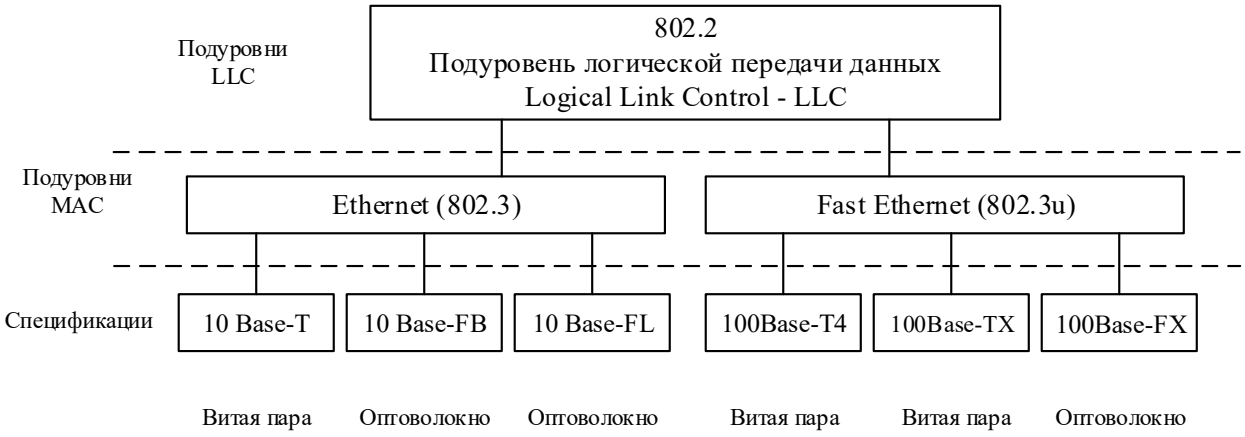


Рисунок 2.27. – Подуровки канального уровня

Международным институтом инженеров по электротехнике и радиоэлектронике (Institute of Electrical and Electronics Engineers - IEEE) было разработано семейство стандартов 802.x, которое регламентирует функционирование канального и физического уровней семиуровневой модели ISO/OSI. Ряд этих протоколов являются общими для всех технологий, например стандарт 802.2, другие протоколы (например, 802.3, 802.3u, 802.5) определяют особенности технологий локальных сетей.

Подуровень LLC реализуется программными средствами. На подуровне LLC существует несколько процедур, которые позволяют устанавливать или

не устанавливать связь перед передачей кадров, содержащих данные, восстанавливать или не восстанавливать кадры при их потере или обнаружении ошибок. Подуровень LLC реализует связь с протоколами сетевого уровня, обычно с протоколом IP. Связь с сетевым уровнем и определение логических процедур передачи кадров по сети реализует протокол 802.2. Протокол 802.1 дает общие определения локальных вычислительных сетей, связь с моделью ISO/OSI.

Подуровень MAC определяет особенности доступа к физической среде при использовании различных технологий локальных сетей. Каждой технологии MAC-уровня (каждому протоколу: 802.3, 802.3u, 802.3zи др.) соответствует несколько вариантов спецификаций (протоколов) физического уровня (рис. 2.27). Спецификация технологии MAC-уровня - определяет среду физического уровня и основные параметры передачи данных (скорость передачи, вид среды, узкополосная или широкополосная).

Рассмотрим базовые правила построения многосегментных ЛВС Fast Ethernet

1. Повторители Fast Ethernet делятся на два класса:

• ***класс I*** – поддерживает все виды логического кодирования (4B/5B, 8B/6T) и может иметь порты всех трех типов физического уровня: 100Base-TX, 100Base-T4 и 100Base-FX;

• ***класс II*** – поддерживает только один вид логического кодирования (4B/5B или 8B/6T) и имеет либо все порты 100Base-T4, либо порты 100Base-TX и 100Base-FX, так как последние используют один логический код 4B/5B.

2. Максимальное число повторителей (П) в одном домене коллизий:

• только 1 повторитель класса I из-за большой задержки распространения сигнала – 70 bt, обусловленной необходимостью транслировать различные системы сигнализации (рис.2.28,а);

• 2 повторителя класса II, вносящих меньшую задержку при передаче сигналов: 46 bt для портов 100Base-TX и 100Base-FX и 33,5 bt для портов 100Base-T4 (рис.2.28,б).

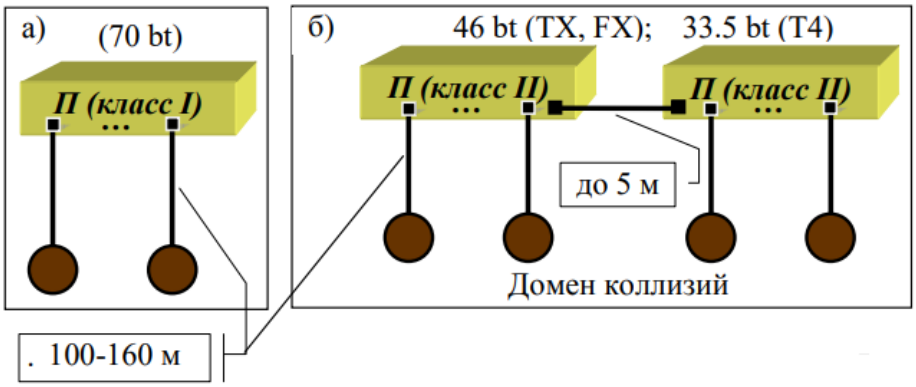


Рисунок 2.28. – Повторители в домене коллизий

3. Максимальное расстояние от повторителя до рабочей станции зависит от типа кабельной системы и составляет 100-160 м.

4. Максимальное расстояние между повторителями класса II – 5 м.

5. Несколько доменов коллизий могут объединяться с помощью коммутаторов и маршрутизаторов, образуя сети произвольных размеров (рис.2.29)

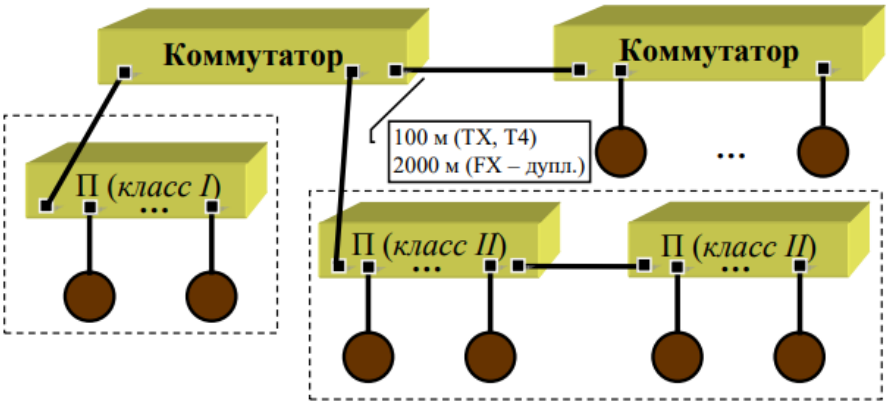


Рисунок 2.29. – Сети произвольных размеров

Таким образом, чтобы сеть Ethernet, состоящая из сегментов, работала корректно, необходимо, чтобы выполнялись три основных условия.

1 Количество станций в сети не должно превышать 1024 (с учетом ограничений для коаксиальных сегментов).

2 Удвоенная ***задержка распространения сигнала*** (Path Delay Value, PDV) между двумя самыми удаленными друг от друга станциями сети не превышает 575 битовых интервалов.

3 Сокращение ***межкадрового расстояния*** (Interpacket Gap Shrinkage) при прохождении последовательности кадров через все повторители не более чем на 49 битовых интервалов (напомним, что при отправке кадров станция обеспечивает начальное межкадровое расстояние в 96 битовых интервалов).

Соблюдение этих требований обеспечивает корректность работы сети даже в случаях, когда нарушаются простые правила конфигурирования, определяющие максимальное количество повторителей и максимальную длину сегментов каждого типа.

Требование на минимальное межкадровое расстояние связано с тем, что при прохождении кадра через повторитель это расстояние уменьшается.

Каждый пакет, принимаемый повторителем, ресинхронизируется для исключения дрожания сигналов, накопленного при прохождении последовательности импульсов по кабелю и через интерфейсные схемы.

Процесс ресинхронизации обычно увеличивает длину преамбулы, что уменьшает межкадровый интервал.

При прохождении кадров через несколько повторителей межкадровый интервал может уменьшиться настолько, что сетевым адаптерам в последнем сегменте не хватит времени на обработку предыдущего кадра, в результате чего кадр будет просто потерян. Поэтому не допускается суммарное уменьшение межкадрового интервала более чем на 49 битовых интервалов.

Величину уменьшения межкадрового расстояния при переходе между соседними сегментами обычно называют в англоязычной литературе ***Segment Variability Value, SVV***, а суммарную величину уменьшения межкадрового интервала при прохождении всех повторителей - ***Path Variability Value, PVV***. Очевидно, что величина PVV равна сумме SVV всех сегментов, кроме последнего.

Для упрощения расчетов обычно используются справочные данные, содержащие значения задержек распространения сигналов в повторителях, приемопередатчиках и в различных физических средах.

В таблице 2.1 приведены данные, необходимые для расчета значения PDV для всех физических стандартов сетей Ethernet, взятые из справочника Technical Reference Pocket Guide (Volume 4, Number 4) компании Bay Networks.

Таблица 2.1. - – Значения задержек распространения сигналов

Тип сегмента	База левого сегмента	База промежу-точного сегмента	База правого сегмента	Задержка среды на 1 м	Максимальная длина сегмента
10Base-5	11.8	46.5	169.5	0.0866	
10Base-2	11.8	46.5	169.5	0.1026	
10Base-T	15.3	42.0	165.0	0.113	
10Base-FB	-	24.0	-	0.1	
10Base-FL	12.3	33.5	156.5	0.1	
FOIRL	7.8	29.0	152.0	0.1	

Одним из наиболее часто критикуемых ограничений Fast Ethernet является диаметр сети, который не должен превышать 205 метров. Такое ограничение затрудняет прямую замену некоторых сетей Ethernet на Fast Ethernet. Поставщики других технологий, в частности Token Ring, 100 VG AnyLAN и FDDI, подчеркивают, что их технологии могут поддерживать сети гораздо большего диаметра. Это действительно так и первоначально ограничивало применение Fast Ethernet сетями рабочих групп и подразделений. Тем не менее такое ограничение топологии может быть легко

преодолено путем использования переключателей и полнодуплексных волоконно-оптических связей.

Способом ***преодоления ограничений топологии*** является разбиение единой области коллизий на несколько при помощи переключателя. Диаметр сети Fast Ethernet, использующей медный кабель и повторитель Класса I, не может превысить 200 метров. Если мы добавим к этой сети единственный переключатель и установим повторители на различные порты, то максимальный диаметр полной переключаемой ЛВС возрастет до 400 метров.

Реальное преимущество сети с переключателями проявляется тогда, когда несколько переключателей соединяются полнодуплексным волоконно-оптическим кабелем, длина которого может достигать 2000 метров (в случае применения многомодового кабеля. При применении одномодового кабеля расстояния достигают десятков километров и зависят от типа используемого оборудования). Этот прием прекрасно подходит для опорной сети.

Таким образом при проектировании сети, особое внимание надо уделить и выбору соответствующего оборудования, учитывая множество факторов, в том числе:

- уровень стандартизации оборудования и его совместимость с наиболее распространенными программными средствами;
- скорость передачи информации и возможность ее дальнейшего увеличения;
- возможные топологии сети и их комбинации (шина, пассивная звезда, пассивное дерево);
- метод управления обменом в сети (CSMA/CD, полный дуплекс или маркерный метод);
- разрешенные типы кабеля сети, его максимальную длину, защищенность от помех;
- стоимость и технические характеристики конкретных аппаратных средств (сетевых адаптеров, трансиверов, репитеров, концентраторов, коммутаторов).

3. Объединения сетей и глобальные сети

3.1. Принципы межсетевого взаимодействия

Межсетевое взаимодействие — это способ соединения компьютерной сети с другими сетями с помощью шлюзов, которые обеспечивают общепринятый порядок маршрутизации пакетов информации между сетями. Полученная система взаимосвязанных сетей называется составной сетью, или просто интерсетью.

Наиболее ярким примером межсетевого взаимодействия является Интернет, в полном смысле слова – СЕТЬ СЕТЕЙ, основанная на многих базовых технологиях оборудования, но объединённая стандартным набором протоколов межсетевого взаимодействия, известного как ***TCP/IP***.

Самым простым примером составной сети являются две локальные сети, соединённые с помощью маршрутизатора. Использование коммутатора или концентратора для соединения локальных сетей не предполагает межсетевого взаимодействия, а лишь расширяет первоначальную сеть.

Межсетевое взаимодействие разрабатывалось как способ соединения совершенно различных типов сетевых технологий, но стало популярно благодаря растущей потребности в соединении двух или более локальных сетей через некую глобальную сеть. Оригинальный термин для межсетевого взаимодействия был - ***CATENET***.

Межсетевое взаимодействие включает в себя соединение других типов компьютерных сетей, таких как персональные вычислительные сети. Сетевые элементы для соединения отдельных сетей в сети ***ARPANET*** (предшественник Интернета), первоначально назывались шлюзами, но термин устарел в этом контексте из-за возможности путаницы между названиями функционально разных устройств. Сегодня смежные шлюзы называются ***маршрутизаторами***.

Другой тип межсетевой связи сетей обычно используется в пределах предприятия на канальном уровне сетевой модели взаимодействия открытых систем - OSI, то есть на аппаратноориентированном уровне стека протоколов TCP/IP. Такая взаимосвязь осуществляется с помощью ***сетевых мостов*** (компьютерное сетевое устройство, которое создает единую совокупную сеть из нескольких сетей связи или сегментов сети.) и ***коммутаторов*** (устройство, предназначенное для соединения нескольких узлов компьютерной сети в пределах одного или нескольких сегментов сети), а сам тип взаимодействия иногда неправильно называют ***межсетевым***, но получившаяся система — это просто более крупная единая подсеть, и не требуется никакого межсетевого

протокола, как например IP для работы таких сетей.

Однако, любая компьютерная сеть может быть преобразована в интерсеть путём деления сети на сегменты и логического разделения трафика на сегменты маршрутизатором.

Протокол IP предназначен для обеспечения ненадёжного (не гарантированного) обслуживания пакетов по всей сети. Такая архитектура позволяет избежать промежуточных элементов сети для поддержания любого состояния сети. Напротив, эта функция возложена на конечные точки каждого сеанса связи.

Для надёжной передачи данных приложения должны использовать соответствующие протоколы транспортного уровня, например - протокол управления передачей (***TCP***), который обеспечивает надёжный поток данных. Некоторые приложения для задач, которые не требуют надёжной доставки данных или требуют доставки пакетов в режиме реального времени, таких как потоковое видео или голосовой чат, используют более простой транспортный протокол без установления соединения, протокол пользовательских Дейтаграмм (***UDP***).

Для описания протоколов и методов, используемых в межсетевом взаимодействии, используются две архитектурные модели.

Сложную, структурированную ЛВС, интегрирующую различные базовые технологии, можно создать средствами не только сетевого уровня, но и канального. Для этого надо использовать некоторые типы мостов и коммутаторов. Мост или коммутатор разделяет сеть на сегменты, локализуя трафик внутри сегмента. При этом сеть разделяют на отдельные подсети, из которых могут быть построены составные сети достаточно крупных размеров. Однако построение сложных сетей только на основе оборудования канального уровня (маршрутизаторов и коммутаторов, выполняющих функции повторителей, мостов и коммутаторов) имеет существенные ограничения и недостатки. В топологии такой сети должны отсутствовать петли, логические сегменты сети слабо изолированы друг от друга, достаточно сложно решается задача управления трафиком, система адресации одно-уровневая и недостаточно гибкая. Возможностью трансляции протоколов канального уров-ня обладают далеко не все типы мостов и коммутаторов, к тому же эти возможности огра-ничены. Естественное решение - это привлечение средств более высокого сетевого уровня.

Основная идея введения сетевого уровня при построении больших сетей состоит в следующем. Сеть рассматривают как совокупность нескольких и называют ***составной сетью*** или ***интерсетью*** (***Internet***).

Сети, входящие в составную сеть, называют подсетями, составляющими

сетями или просто сетями. Когда две или более сети организуют совмест-ную транспортную службу, то режим взаимодействия называют межсетевым взаимодействием (***Internet working***).

Подсети соединяются между собой маршрутизаторами, в качестве которых могут использоваться как собственно маршрутизаторы, так и коммутаторы. Компонентами составной сети могут быть как локальные, так и глобальные сети. Внутренняя структура каждой сети не имеет значения при рассмотрении сетевого протокола. Все узлы в пределах одной подсети взаимодействуют, используя единую для них технологию. Так, в составную сеть может входить несколько сетей разных технологий: локальные сети ***Ethernet***. ***Fast Ethernet***, ***Token Ring***, ***FDDI*** и глобальные сети ***Frame Relay***, ***X.25***, ***ISDN***, ***ATM***. Каждая из этих технологий достаточна, чтобы организовать взаимодействие всех узлов в своей подсети, но не обеспечивает организацию связи между произвольно выбранными узлами, принадлежащими разным подсетям. Для организации взаимодействия требуются дополнительные средства, которые и предоставляет сетевой уровень.

Сетевой уровень обеспечивает работу всех подсетей в процессе передачи пакетов сообщений по составной сети.

Для перемещения данных в пределах подсетей сетевой уровень обращается к используемым в этих подсетях технологиям. Для сетевого уровня предусматривается собственная система адресации, не зависящая от способов адресации узлов в отдельных подсетях, которая позволяет на сетевом уровне универсальным и однозначным способом идентифицировать любой узел объединенной или составной сети.

Естественным ***способом формирования сетевого адреса*** является уникальная нумерация всех подсетей составной сети и нумерация всех узлов в пределах каждой подсети. Таким образом, сетевой адрес включает в себя номера сети (подсети) и узла. В качестве номера узла может быть задано некоторое число, никак не связанное с локальной технологией, которое однозначно идентифицирует узел в пределах данной подсети. Такой подход более универсален и характерен для стека протоколов TCP/IP: каждый узел составной сети имеет наряду с локальным адресом еще один - универсальный сетевой.

Данные, которые поступают на сетевой уровень и которые необходимо передать через составную сеть, снабжаются заголовком сетевого уровня. Данные вместе с заголовком образуют пакет. Заголовок пакета сетевого уровня имеет унифицированный формат, не зависящий от форматов кадров канального уровня тех сетей, которые могут входить в объединенную сеть, он, наряду с другой служебной информацией, включает данные о номере сети,

которой предназначается этот пакет. Сетевой уровень определяет маршрут и перемещает пакет между подсетями.

При передаче из одной подсети в другую пакет сетевого уровня, инкапсулированный в прибывший канальный кадр первой подсети, освобождается от заголовков этого кадра и окружается заголовками кадра канального уровня следующей подсети. Информацией, на основе которой делается эта замена, являются служебные поля пакета сетевого уровня. В поле адреса назначения нового кадра указывается локальный адрес следующего маршрутизатора. При этом если в подсети данные доставляются средствами канального и физического уровней, то пакеты сетевого уровня упаковываются в кадры канального. Если же в какой-либо подсети для транспортировки сообщений используется технология, основанная на стеках с большим числом уровней, то пакеты сетевого уровня упаковываются в блоки передаваемых данных самого высокого уровня подсети.

Кроме номера сети заголовок сетевого уровня должен содержать и другую информацию, необходимую для успешного перехода пакета из одной сети в другую. К такой информации может относиться, например, номер фрагмента пакета, необходимый для успешного проведения операций сборки-разборки фрагментов при соединении сетей с разными максимальными размерами пакетов, время жизни пакета, указывающее, как долго он перемещается по сети, качество услуги - критерий выбора маршрута при межсетевых передачах.

Организация межсетевого взаимодействия средствами сетевого уровня эталонной модели ВОС, в основе которого лежит организация совместной или единой транспортной службы для всей составной или единой транспортной сети, служит основой для интеграции различных сетевых технологий в современных цифровых сетях.

3.2. Сети TCP/IP

В стандартной модели взаимодействия открытых систем в функции сетевого уровня входит решение следующих задач:

- передача пакетов между конечными узлами в составных сетях;
- выбор маршрута передачи пакетов, наилучшего по некоторому критерию;
- согласование разных протоколов канального уровня, использующихся в отдельных подсетях одной составной сети.

Сетевой уровень (протокол IP) мультиплексирует протокольные блоки транспортного уровня в IPпотоки; при этом сегменты транспортного уровня

могут фрагментироваться (если они превышают максимально допустимый размер, определяемый канальным протоколом). Протокольные блоки IP обычно называют пакетами.

После вычисления маршрута передачи пакета, посредством протокола ARP определяется физический адрес следующего на маршруте хоста и пакет направляется на физический уровень сети. Иногда бывает необходимо решить обратную задачу, то есть по заданному физическому адресу определить логический сетевой адрес устройства. В частности, эта проблема актуальна для процедуры загрузки бездисковых станций, когда такая станция рассылает в широковещательном режиме запрос, содержащий ее физический адрес и «просьбу» сообщить ей логический сетевой адрес. Этот запрос обрабатывается сервером RARP, который и передает пославшей его станции требуемую информацию.

Физический уровень TCP/IP сети может использовать любую технологию канального уровня – Ethernet, Token Ring, ATM, PPP и т.д. Но для обеспечения прозрачности физического уровня необходимо, чтобы на сетевом уровне были предусмотрены процедуры дефрагментации пакетов до размера, разрешенного соответствующим протоколом канального уровня. Обратная процедура, т.е. объединение пакетов малых размеров до величины, приемлемой на канальном уровне, не предусматривается.

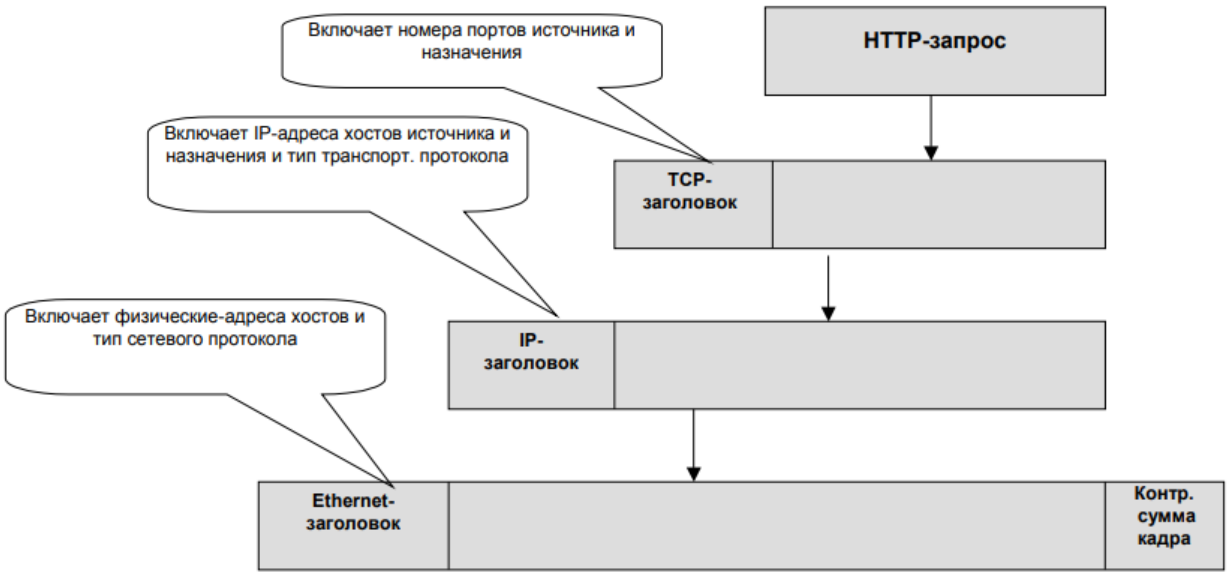


Рисунок 3.1. – Инкапсуляция протокольных блоков в TCP/IP стеке

Протокольные блоки вышележащих уровней инкапсулируются в протокольные блоки нижележащих уровней так, как это показано на слайде. При этом, блок каждого уровня содержит специфическую информацию, позволяющую точно адресовать его. Так, сегмент TCP (UDP

Основной протокол стека TCP/IP, - Internet Protocol (IP), - по своим

функциям соответствует сетевому уровню модели взаимодействия открытых систем (OSI). Механизмы протокола, описанные в документе RFC 791, обеспечивают ненадежную доставку пакетов данных между сетевыми устройствами (устройствами, имеющими сетевой адрес) в режиме без предварительного установления соединения (дейтограммный сервис). Этот тип сервиса часто называют сервисом «настолько хорошо, как получиться» (best effort service), что отражает отсутствие в протоколе процедур контроля доставки пакетов. Решение задачи надежности доставки возлагается на протоколы верхних уровней, главным образом на TCP. Основными функциями протокола IP являются:

- формирование пакетов из сегментов транспортного уровня, с предварительной фрагментацией (если необходимо) последних;
- обеспечение логической адресации сетевых устройств;
- поддержка процесса маршрутизации;
- продвижение пакетов от одного узла коммутации до другого.

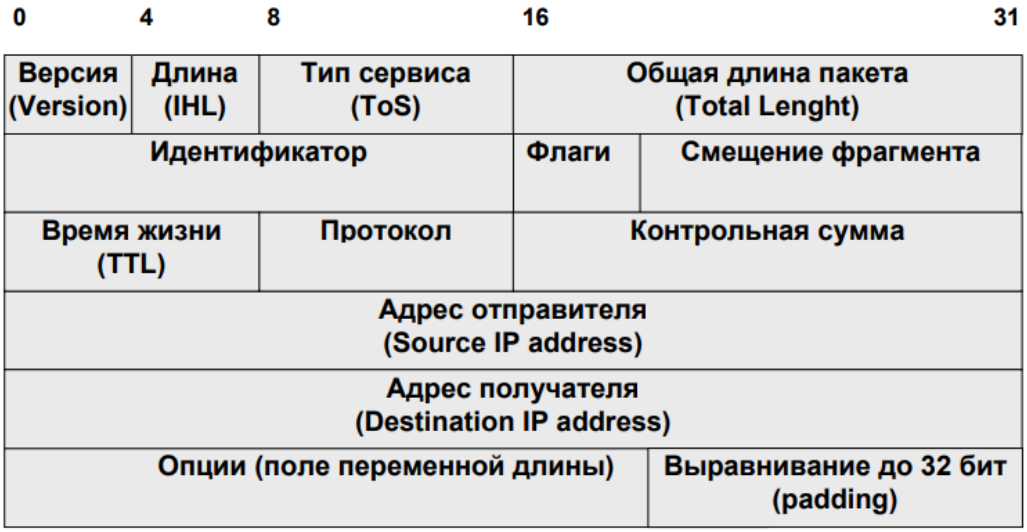


Рисунок 3.2. – Формат заголовка IP-пакета

Версия (Version) – поле определяет номер версии протокола. В настоящее время используется версия 4 и ведется активная подготовка к переходу на версию 6. Версия 5 описывает протокол ST2, разработанный для передачи данных потоковых приложений реального времени. Поле проверяется перед обработкой пакета и пакеты несогласующейся с протокольным стеком приемника версией, отбрасываются. Одновременно, включение версии в каждую дейтограмму позволяет использовать разные, но согласующиеся, версии на разных хостах.

Длина заголовка (Internet Header Length, IHL) - Поле определяет длину заголовка, измеренную в 32-битных словах. Корректный заголовок имеет

длину не менее 5 слов. Длина поля опций (в 32-разрядных словах) может быть определена как значение этого поля минус 5.

Тип сервиса (Type of service, ToS) – определяет тип требуемого обслуживания пакета. Первые три бита задают уровень приоритета обслуживания (0-7), 3-5 биты определяют требования к задержке (какая получится, низкая), уровень пропускной способности (обычный, высокий) и надежность доставки (какая получится, высокая). Практически, большая часть маршрутизаторов игнорирует данные этого поля. Однако в настоящее время в связи с разработкой механизмов обеспечения в IP-сетях служб с гаранированным качеством обслуживания делаются попытки использования значений этого поля.

Общая длина (Total length) - поле содержит общую длину пакета, размер которого не может превышать 65535 байт. Практически пакеты такой длины никогда не используются, поскольку технологии канального уровня накладывают свои ограничения. Так, Ethernet не допускает кадров с длиной более 1500 байт, FDDI – 4096 байт и т.д. В этой связи, протокол IP выполняет фрагментацию сегментов данных, поступающих к нему от TCP и UDP протоколов. Следует отметить, что маршрутизатор не выполняет сборку пакетов, даже если следующая сеть имеет параметр MTU (Maximum Transmission Unit), допускающий более крупные пакеты. Сборка пакетов в исходный сегмент производится на месте назначения.

Поля ***«Идентификатор»***, ***«Флаги»*** и ***«Смещение фрагмента»*** управляют процессом сборки сегмента.

Время жизни (Time to live, TTL) - поле, определяющее максимальное время, которое пакет может существовать в сети. Значение этого поля (в секундах) устанавливается при отправке пакета и уменьшается на единицу по мере прохождения им маршрутизаторов. При достижении нулевого значения этого поля пакет уничтожается. Максимальное значение поля – 255 секунд. Этот механизм помогает избежать перегрузок сети при возникновении ошибок в таблицах маршруизации, приводящих к образованию петель.

Протокол (Protocol) – поле указывает модулю какого протокола (TCP, UDP, ICMP) передать полученный IP-пакет. На рисунке 3.3. приведены значения этого поля для некоторых из протоколов. В дальнейшем нас будут интересовать TCP, UDP, ICMP.

Значение поля	Ключевое слово	Протокол
0	Reserved	Зарезервировано
1	ICMP	Internet Control Message Protocol
2	IGMP	Internet Group Management Protocol
4	IP	Internet Protocol
6	TCP	Transmission Control Protocol
17	UDP	User Datagram Protocol
89	OSPF	Open Shortest Path First

Рисунок 3.3. – Поле «Протокол» заголовка IP-пакета

Контрольная сумма (Header checksum) – поле содержит значение контрольной суммы, рассчитанной только по заголовку. Поскольку значения некоторых полей заголовка изменяются по мере прохождения пакета по маршруту (поле TTL, например), то значения рассматриваемого поля проверяются и пересчитываются на каждом маршрутизаторе. Этот механизм является единственным средством обеспечения достоверности передачи, содержащимся в протоколе IP.

Адрес отправителя (Source IP address) и ***Адрес получателя (Destination IP address)*** – поля одинаковой длины (32 бита), содержащие соответствующие адреса. Правила адресации в IP-сетях будут рассмотрены далее.

Опции (Options) – необязательное поле, используемое при отладке сетей и для запроса определенных специфических процедур обработки. В настоящее время используется крайне редко. В связи с разработкой новых протоколов, обеспечивающих большую гибкость в обработке IP-трафика, возможность использования этих полей вновь стала предметом обсуждения комитетов по стандартизации.

Для идентификации каждого компьютера в IP-сети необходима система их адресации. При этом учитывается, что сетевые устройства (компьютер, маршрутизатор и т.д.) могут иметь несколько сетевых интерфейсов, и каждый из них должен иметь уникальный адрес.

Классификация IP адресов

Класс сетей	Значение первого байта адреса	Количество сетей	Количество хостов в сети класса	Удельный вес класса в адресном пространстве, %
A	001 – 126	126	16 777 214	50
B	128 – 191	16 384	65 534	25
C	192 - 223	2 097 152	254	12,5

Рисунок 3.4. – Адресация в сетях IP

IP -адрес строится по двухуровневой иерархии, т.е. он объединяет в себе адрес сети и адрес хоста. Разделение сетевого адреса на 2 части имеет большой практический смысл, ибо позволяет магистральным маршрутизаторам существенно сократить размер своих таблиц коммутации, формируя их на основании только сетевой части адреса назначения. Для удовлетворения потребностей адресации сетей различного масштаба были введены несколько классов сетей, отличающиеся размером полей, отводимых для указания номера сети и номера хоста. При этом, размер поля полного адреса всегда равен 32 битам.

IP-адрес обычно записывается в форме 4-х трехразрядных десятичных чисел, разделенных точкой. Каждое из этих десятичных чисел соответствует одному байту двоичного представления адреса. Так, например, адрес 10000000 10000111 01000100 00000101 в десятичном представлении имеет вид 128.135.68.5. В этом случае, т.к. первые два бита адреса – 10, то это адрес хоста, принадлежащего сети класса B и, следовательно, левые 16 бит являются адресом сети, а правые16 бит – адресом хоста.

Некоторые адреса являются зарезервированными и не могут присваиваться хостам. Так, адрес 127.x.x.x (x – означает любое число, обычно

0) зарезервирован для обратной связи, используемой при тестировании взаимодействия процессов на одной сетевой станции. Когда приложение использует этот адрес в качестве адреса назначения, стек TCP/IP данного хоста возвращает данные приложению, ничего не передавая на физический интерфейс. Поэтому адреса, начинающиеся на 127, запрещается присваивать сетевым устройствам. Другим зарезервированным адресом является, так называемый, широковещательный адрес, содержащий 1, или 0, во всех своих битах. Пакет с адресом назначения 255.255.255.255 (1.1.1.1) будет доставлен всем устройствам сети, к которой принадлежит узел-отправитель, но маршрутизаторы такие пакеты не обрабатывают. Существует и направленное широковещание – способ адресации, при котором один пакет, отосланный в определенную сеть, будет доставлен всем ее хостам. Такой пакет должен содержать корректный адрес сети и иметь все биты адреса хоста равными 1.

Каждый хост (станция, маршрутизатор) ведет свои маршрутные таблицы, которые и определяют порядок обработки IP-пакетов. Передача пакетов между конечными станциями, требует взаимодействия IP-модулей программного обеспечения этих станций и маршрутизаторов, связывающих сети, в которых они (станции) находятся. Рассмотрим, как обрабатывается пакет на этих сетевых устройствах.

Если в таблице маршрутизации станции-отправителя указано, что станция назначения является непосредственно присоединенной к той же ЛВС, то из таблицы физических адресов, которая ведется на каждой сетевой станции, извлекается физический адрес узла назначения, пакет ***инкансулируется*** (упаковываются данные в один компонент) в кадр канального протокола и передается к станции назначения. Если таблица маршрутизации станции-отправителя не содержит искомый сетевой адрес, то пакет отправляется по адресу маршрутизатора, который был указан при конфигурировании станции в качестве шлюза по умолчанию (***default router, default gateway***). Этот шлюз обязательно имеет физический интерфейс в той же ЛВС, что станция-отправитель. При получении пакета маршрутизатор проверяет, не совпадает ли адрес назначения этого пакета с его собственным IP-адресом. Если это так, то пакет передается модулю протокола, указанного в поле «Протокол» заголовка пакета. В противном случае, маршрутизатор посредством своей таблицы определяет адрес следующего хоста, которому он должен передать этот пакет, и свой интерфейс, на который следует его направить.

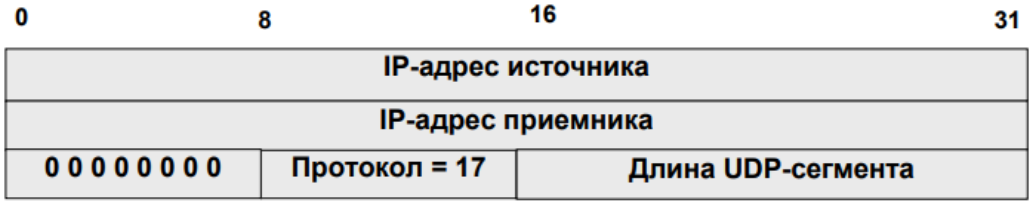
Каждая строка в таблице маршрутизации содержит следующую информацию: IP-адрес сети (узла) назначения, IP-адрес следующего маршрутизатора, способного обеспечить передачу пакета в эту сеть (этому

узлу), имя выходного интерфейса и некоторые флаги. Флаги содержат уточняющую информацию о каждой записи в таблице.

Два протокола транспортного уровня, UDP и TCP, обеспечивают IP-сетям механизмы взаимодействия прикладных процессов, выполняющихся на конечных станциях. Протокол IP «умеет» доставлять пакеты данных взаимодействующим хостам, но не «знает» как обеспечить взаимосвязь приложений и не имеет почти никаких средств обеспечения надежности доставки сообщений - он проверяет лишь целостность заголовка пакета.

Протокол UDP (User Datagram Protocol) описан в документе RFC 768. Он ориентирован на сервис без установления соединений и не обеспечивает надежную передачу сегментов между сетевыми приложениями. Это очень простой протокол, который развивает возможности IP-протокола лишь в части демультиплексирования потока пакетов по признаку принадлежности их определенному приложению и контроля целостности данных. Взаимодействие между прикладными процессами UDP реализует посредством механизма протокольных портов. Протокольный порт можно определить как абстрактную точку присутствия конкретной прикладной программы, выполняющейся на конкретном хосте. Когда рабочая станция получает пакет, в котором указан ее IP-адрес, она может направить его определенной программе, используя уникальный номер порта, назначенный этой программе в ходе выполнения процедуры установления соединения. Таким образом, в стеке протоколов TCP/IP порт является механизмом поддержания рабочей станцией одновременного выполнения нескольких прикладных процессов.

Формат UDP-сегмента



Формат псевдозаголовка UDP-сегмента

Рисунок 3.5. – Протокол UDP

Каждый порт (прикладной процесс) идентифицируется целым положительным числом (номером порта). Номера портов приложения,

выполняющегося на разных станциях, указываются в заголовке UDP-сегмента. Эта информация дополняется на сетевом уровне IP-адресами взаимодействующих станций. Благодаря этому, создается видимость непосредственного обмена данными между процессами.

Сегмент данных протокола UDP (иногда его называют пользовательской дейтограммой) состоит из двух частей: заголовка и области данных (рис. 3.5). Заголовок имеет четыре 16-битных поля, определяющих порт отправителя, порт получателя, длину сегмента и контрольную сумму.

Поле «Длина UDP-сегмента» содержит количество байтов в дейтограмме с учетом длины ее заголовка.

Вычисление контрольной суммы дейтограммы UDP является опциональным. При работе в надежных локальных сетях она не вычисляется и тогда это поле заполняется нулями. Процедура подсчета контрольной суммы содержит две особенности. Первая состоит в дополнении дейтограммы нулевыми битами до размера, кратного 16. Это делается только на время вычисления контрольной суммы, и незначащие нули не передаются. Второй особенностью является дополнение, на период подсчета контрольной суммы, заголовка сегмента ***псевдозаголовком***

Псевдозаголовок включается перед заголовком дейтограммы; он имеет длину 12 байтов; поле «Протокол» содержит тип протокола сетевого уровня (IP, ICMP); его значение, как и значения полей с IP-адресами, должны быть извлечены из заголовка IP пакета. Поле «Контрольная сумма» на время ее вычисления заполняется нулями.

Такое дополнение дейтограммы выполняется как на передающей, так и на приемной станции и оно служит гарантией, что если контрольные суммы совпали, то дейтограмма достигла нужной станции и нужного порта. Еще раз подчеркну, что псевдозаголовок и дополнение нулями не передаются.

Если контрольные сумма, вычисленная приемником, не совпала с контрольной суммой, указанной в дейтограмме, то UDP-сегмент уничтожается и никаких уведомлений передающей станции об этом не передается.

Рассмотрим дополнительные протоколы ***DHCP***, ***ARP***, ***RARP***, ***ICMP*** которые являются вспомогательными или сопутствующими протоколами стека протоколов TCP/IP и протоколы OSPF, RIP которые относятся к протоколам динамической маршрутизации.

Для облегчения работы администраторов и был создан протокол ***DHCP*** (***Dynamic Host Configuration Protocol***).

Протокол DHCP работает в соответствии с моделью клиент-сервер. Во время старта системы компьютер, являющийся DHCP-клиентом, посылает в

сеть широковещательный запрос на получение IP-адреса. DHCP-сервер откликается и посылает сообщение-ответ, содержащее IP-адрес и некоторые другие конфигурационные параметры.

Хотя основным назначением DHCP является динамическое назначение IP-адресов, но данный протокол может поддерживать ручное назначение адресов.

При этом сервер DHCP может работать в разных режимах:

1) ручное назначение статических адресов;

2) автоматическое назначение статических адресов;

3) автоматическое распределение динамических адресов.

Во всех режимах работы администратор при конфигурировании DHCP-сервера сообщает ему один или несколько диапазонов IP-адресов, причем все эти адреса относятся к одной сети, то есть имеют одно и то же значение в поле номера сети.

1) ***В ручном режиме*** администратор, помимо пула доступных адресов, снабжает DHCP-сервер информацией о жестком соответствии IP-адресов физическим адресам или другим идентификаторам клиентских узлов. DHCP-сервер, пользуясь этой информацией, всегда выдает определенному DHCP-клиенту один и тот же назначенный ему администратором IP-адрес (а также набор других конфигурационных параметров.

2) ***В режиме автоматического*** назначения статических адресов DHCP сервер самостоятельно без вмешательства администратора произвольным образом выбирает клиенту IP-адрес из пула наличных IP-адресов. Адрес дается клиенту из пула в постоянное пользование, то есть между идентифицирующей информацией клиента и его IP-адресом по-прежнему, как и при ручном назначении, существует постоянное соответствие. Оно устанавливается в момент первого назначения DHCP-сервером IP-адреса клиенту. При всех последующих запросах сервер возвращает клиенту тот же самый IP-адрес.

3) ***При динамическом распределении адресов*** DHCP-сервер выдает адрес клиенту на ограниченное время, называемое сроком аренды. Когда компьютер, DHCP-клиент, удаляется из подсети, назначенный ему IP-адрес автоматически освобождается. Когда компьютер подключается к другой подсети, то ему автоматически назначается новый адрес.

Администратор управляет процессом конфигурирования сети, определяя два основных параметра конфигурации DHCP-сервера: пул адресов, доступных распределению, и срок аренды. Срок аренды диктует, как долго компьютер может использовать назначенный IP-адрес, перед тем как снова запросить его от DHCP-сервера. Срок аренды зависит от режима работы

пользователей сети. Если это небольшая сеть учебного заведения, куда со своими компьютерами приходят многочисленные студенты для выполнения лабораторных работ, то срок аренды может быть равен длительности лабораторной работы. Если же это корпоративная сеть, в которой сотрудники предприятия работают на регулярной основе, то срок аренды может быть достаточно длительным — несколько дней или даже недель.

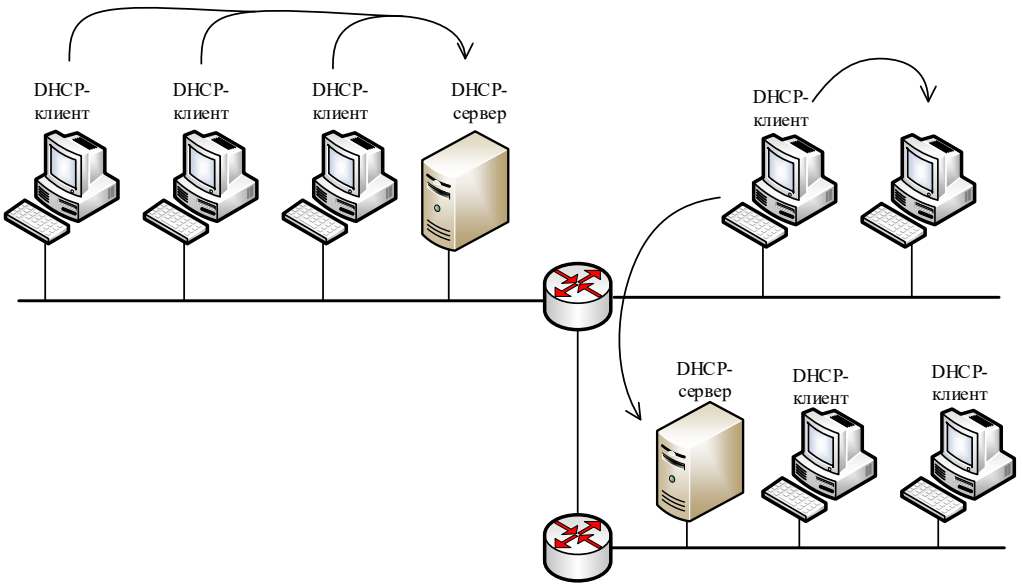


Рисунок 3.6. – Схемы взаимного расположение серверов и клиентов DHCP

DHCP-сервер должен находиться в одной подсети с клиентами, учитывая, что клиенты посылают ему широковещательные запросы. Для снижения риска выхода сети из строя из-за отказа DHCP-сервера в сети иногда ставят резервный DHCP-сервер.

Когда обращаемся к ресурсу в локальной сети или же ресурсу за её пределами, вводим сетевой адрес, срабатывает протокол ***ARP (Address Resolution Protocol)***, задача которого – преобразовать IP адрес в адрес локальной сети (для технологий Ethernet, Toking Ring, FDDI – это MAC адрес).

В локальных сетях протокол ARP использует широковещательные кадры протокола канального уровня для поиска в сети узла с заданным IP-адресом.

Узел (компьютер, маршрутизатор и др.), которому нужно выполнить отображение IP-адреса на локальный адрес (на MAC адрес), формирует ARP запрос (рис. 3.7), вкладывает его в кадр протокола канального уровня (рис. 3.8), указывая в нем известный IP-адрес, и рассылает запрос широковещательно.

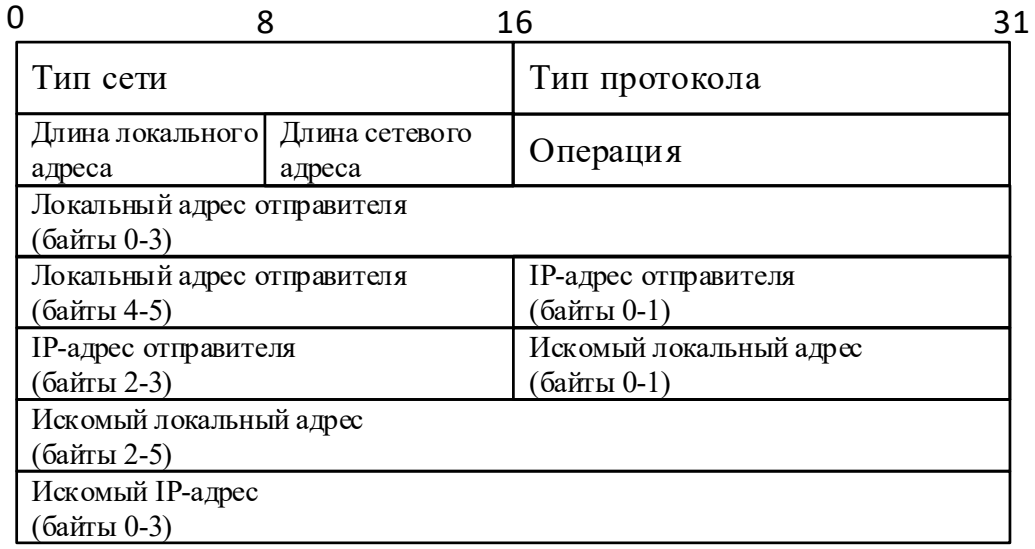


Рисунок 3.7. – Формат пакета протокола ARP стека TCP/IP и канального Ethernet

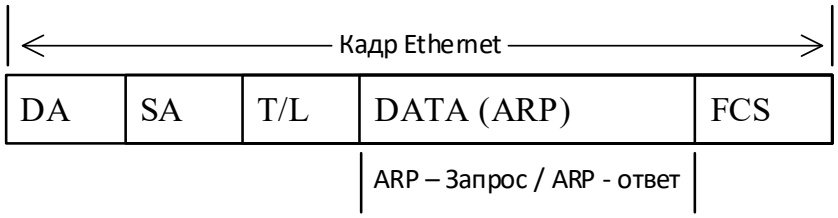


Рисунок 3.8. – Инкапсуляция сообщений ARP в кадр Ethernet

Все узлы локальной сети получают ARP запрос и сравнивают указанный там IP-адрес с собственным. В случае их совпадения узел формирует ARP-ответ, в котором указывает свой IP-адрес и свой локальный адрес и отправляет его уже направленно, так как в ARP запросе отправитель указывает свой локальный адрес. ARP-запросы и ответы используют один и тот же формат пакета. Так как локальные адреса могут в различных типах сетей иметь различную длину, то формат пакета протокола ARP зависит от типа сети. На рисунке 14.3 показан формат пакета протокола ARP для передачи по сети Ethernet.

В поле типа сети для сетей Ethernet указывается значение 1. Поле типа протокола позволяет использовать пакеты ARP не только для протокола IP, но и для других сетевых протоколов. Для IP значение этого поля равно 0x0800.

Длина локального адреса для протокола Ethernet равна 6 байтам, а длина IPадреса - 4 байтам. В поле операции для ARP запросов указывается значение 1 для протокола ARP и 2 для протокола RARP.

Узел, отправляющий ARP-запрос, заполняет в пакете все поля, кроме поля искомого локального адреса (для ARP-запроса не указывается искомый IPадрес). Значение этого поля заполняется узлом, опознавшим свой IP-адрес.

RARP (Reverse Address Resolution Protocol — обратный протокол преобразования адресов) — протокол третьего (сетевого) уровня модели OSI — выполняет обратное отображение адресов, то есть преобразует аппаратный (MAC) адрес в IP-адрес. Протокол применяется во время загрузки узла (например, компьютера), когда он посылает групповое сообщение-запрос со своим физическим адресом. Сервер принимает это сообщение и просматривает свои таблицы (либо перенаправляет запрос куда-либо еще) в поисках соответствующего физическому IP-адреса. После обнаружения найденный адрес отсылается обратно на запросивший его узел. Другие станции также могут "слышать" этот диалог и локально сохранить эту информацию в своих ARPтаблицах. RARP позволяет разделять IP-адреса между не часто используемыми хост-узлами. После использования каким-либо узлом IP-адреса он может быть освобожден и выдан другому узлу. RARP используется, например, при старте бездисковых станций, не знающих в начальный момент времени своего IPадреса, но знающих MAC-адрес своего сетевого адаптера.

Протокол обмена управляющими сообщениями ICMP (Internet Control Message Protocol) позволяет маршрутизатору сообщить конечному узлу об ошибках, с которыми маршрутизатор столкнулся при передаче какого-либо IP-пакета от данного конечного узла. Управляющие сообщения ICMP не могут направляться промежуточному маршрутизатору, который участвовал в передаче пакета, с которым возникли проблемы, так как для такой посылки нет адресной информации - пакет несет в себе только адрес источника и адрес назначения, не фиксируя адреса промежуточных маршрутизаторов. Протокол ICMP - это протокол сообщения об ошибках, а не протокол коррекции ошибок. Конечный узел может предпринять некоторые действия для того, чтобы ошибка больше не возникала, но эти действия протоколом ICMP не регламентируются. Каждое сообщение протокола ICMP передается по сети внутри пакета IP. Пакеты IP с сообщениями ICMP маршрутизируются точно так же, как и любые другие пакеты, без приоритетов, поэтому они также могут теряться. Кроме того, в загруженной сети они могут вызывать дополнительную загрузку маршрутизаторов. Для того, чтобы не вызывать лавины сообщения об ошибках, потери пакетов IP, переносящие сообщения

ICMP об ошибках, не могут порождать новые сообщения ICMP.

Помимо диагностики ICMP также используется для мониторинга сети. Так, в основе популярных утилит для мониторинга IP-сетей ping и tracert лежат ICMPсообщения. С помощью ICMP-сообщений приложение может определить маршрут перемещения данных, оценить работоспособность сети, определить время прохождения данных до заданного узла, сделать запрос о значении маски определенного сетевого интерфейса и т. п.

Все типы ICMP-сообщений могут быть разделены на два класса:

- диагностические сообщения об ошибках;
- информационные сообщения типа запрос/ответ.

ICMP-сообщение инкапсулируется в поле данных IP-пакета

Рисунок 3.9. – Инкапсуляция и формат ICMP-сообщения

Заголовок ICMP состоит из 8 байт; поля заголовка перечислены ниже.

Тип (размером 1 байт) содержит код, определяющий тип сообщения.

Код (размером 1 байт) более тонко дифференцирует тип ошибки.

Контрольная сумма, подсчитанная для всего ICMP-сообщения, занимает 2 байта.

Поле из 4 байт:

1) В сообщениях типа ***запрос/ответ*** это поле содержит 2-байтовые подполя идентификатора и порядкового номера. Числа из этих подполей дублируются из сообщения-запроса в сообщение-ответ. ***Идентификатор*** позволяет узлу-получателю сообщения определить, какому приложению направлен этот ответ, а порядковый номер используется приложением, чтобы связать ответ с соответствующим запросом (учитывая, что одно приложение может выдать несколько однотипных запросов).

2) В сообщениях об ошибке это поле не используется и заполняется нулями.

Наиболее распространенным протоколом, основанным на дистанционно-векторном алгоритме, является протокол ***RIP (Routing Information Protocol)***. Преимуществом протокола RIP является его вычислительная простота, а недостатками - увеличение трафика при периодической рассылке широковещательных пакетов и неоптимальность найденного маршрута.

Алгоритмы состояния связей обеспечивают каждый маршрутизатор информацией, достаточной для построения точного графа связей сети. Все маршрутизаторы работают на основании одинаковых графов, что делает процесс маршрутизации более устойчивым к изменениям конфигурации. Широковещательная рассылка используется здесь только при изменениях состояния связей, что происходит в надежных сетях не так часто. Маршрутизатор периодически обменивается короткими пакетами со своими ближайшими соседями. Этот трафик также широковещательный, но он циркулирует только между соседями и поэтому не так засоряет сеть.

Протоколом, основанным на алгоритме состояния связей, в стеке TCP/IP является протокол ***OSPF***.

Протокол RIP относится к группе протоколов с дистанционно-векторным алгоритмом (DVA).

Для IP имеются две версии RIP — RIP v1 и RIP v2. Протокол RIP v1 не поддерживает масок. Протокол RIP v2 передает информацию о масках сетей, поэтому он в большей степени соответствует требованиям сегодняшнего дня.

Таблица 3.1. Таблица маршрутизации протокола RIP

Сеть	Следующий переход	Метрика	Таймеры	Флаги
153.19.88.0	Route “A”	2	30-180-240	
198.63.35.0	Route “B”	6	30-180-240	
153.19.89.0	Route “C”	1	30-180-240	

Поля таблицы:

Сеть: адрес сети назначения.

Следующий переход: IP-адрес маршрутизатора, который является следующим звеном при перемещении к адресату.

Метрика/Стоимость: Метрика маршрутизации ─ параметр критерий, по которому определяется наиболее предпочтительный маршрут.

Таймер ─ поле на самом деле представляет три разных таймера, используемых RIP. Таймер обновления маршрутов (***routing update timer***) указывает интервал между обновлениями. Обычно RIP отсылает копию своей таблицы маршрутов каждые 30 секунд.

Второй таймер – это тайм-аут для маршрута-таймера удержания (***route timeout***). Если от какой-то конкретной сети обновление маршрутов не получено в течение 180 секунд, то маршрут помечается как недостижимый. Последний таймер – это таймер удаления маршрута (***route removal timer***).

Флаги: в поле хранятся различные необязательные данные (RIP

используется нечасто).

Протокол OSPF (Open Shortest Path Firs) является достаточно современной реализацией алгоритма состояния связей (он принят в 1991 году) и обладает многими особенностями, ориентированными на применение в больших гетерогенных сетях.

OSPF имеет следующие преимущества:

- Высокая скорость сходимости по сравнению с дистанционно-векторными протоколами маршрутизации;
- Поддержка сетевых масок переменной длины (VLSM);
- Оптимальное использование пропускной способности с построением дерева кратчайших путей.

Принцип работы заключается в следующем:

1. После включения маршрутизаторов протокол ищет непосредственно подключенных соседей и устанавливает с ними «дружеские» отношения.

2. Затем они обмениваются друг с другом информацией о подключенных и доступных им сетях. То есть они строят карту сети (топологию сети). Данная карта одинакова на всех маршрутизаторах.

3. На основе полученной информации запускается алгоритм SPF (Shortest Path First, «выбор наилучшего пути»), который рассчитывает оптимальный маршрут к каждой сети. Данный процесс похож на построение дерева, корнем которого является сам маршрутизатор, а ветвями — пути к доступным сетям. Данный процесс, то есть конвергенция, происходит очень быстро.

3.3. Глобальные сети и перспективные сетевые технологии

Под коммутацией понимается технология выбора направления и организация передачи данных в сетях, имеющих несколько альтернативных маршрутов, по которым может производиться обмен информацией между двумя узлами.

Существуют три принципиально различные схемы коммутации абонентов в сетях:

- коммутация каналов (circuit switching);
- коммутация пакетов (packet switching);
- коммутация сообщений (message switching).

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

В первом случае сеть разрешает устанавливать соединение по инициативе пользователя сети. Коммутация выполняется на время сеанса связи, а затем

(по инициативе одного из взаимодействующих пользователей) связь разрывается.

Во втором случае сеть не предоставляет пользователю возможность выполнить динамическую коммутацию с другим произвольным пользователем сети. Вместо этого сеть разрешает паре пользователей заказать соединение на длительный период времени. Соединение устанавливается не пользователями, а персоналом, обслуживающим сеть. Время, на которое устанавливается постоянная коммутация, измеряется обычно несколькими месяцами. Режим постоянной коммутации в сетях с коммутацией каналов часто называется сервисом ***выделенных (dedicated)*** или ***арендуемых (leased)*** каналов.

Примерами сетей, поддерживающих режим динамической коммутации, являются: телефонные сети общего пользования, локальные сети, сети TCP/IP.

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

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).

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

Сети с динамической коммутацией требуют предварительную процедуру установления соединения между абонентами. Для этого в сеть передается адрес вызываемого абонента, который проходит через коммутаторы и настраивает их на последующую передачу данных. Запрос на установление соединения маршрутизируется от одного коммутатора к другому и в конце концов достигает вызываемого абонента.

При ***коммутации пакетов*** все передаваемые пользователем сети сообщения разбиваются в исходном узле на сравнительно небольшие части, называемые пакетами, пакеты обычно тоже могут иметь переменную длину,

от 46 до 1500 байт.

Каждый пакет снабжается заголовком, в котором указывается адресная информация, необходимая для доставки пакета узлу назначения, а также номер пакета, который будет использоваться узлом назначения для сборки сообщения. Пакеты транспортируются в сети как независимые информационные блоки. Коммутаторы сети принимают пакеты от конечных узлов и на основании адресной информации передают их друг другу, а в конечном итоге - узлу назначения.

Коммутаторы пакетной сети отличаются от коммутаторов каналов тем, что они имеют внутреннюю буферную память для временного хранения пакетов, если выходной порт коммутатора в момент принятия пакета занят передачей другого пакета. В этом случае пакет находится некоторое время в очереди пакетов в буферной памяти выходного порта, а когда до него дойдет очередь, то он передается следующему коммутатору.

Такой режим работы сети называется ***дейтаграммным***, и при его использовании коммутатор может изменить маршрут какого-либо пакета в зависимости от состояния сети - работоспособности каналов и других коммутаторов, длины очередей пакетов в соседних коммутаторах и т. п.

Существует и другой режим работы сети - передача пакетов по ***виртуальному каналу*** (***virtual circuit или virtual channel***).

В этом случае перед тем, как начать передачу данных между двумя конечными узлами, должен быть установлен виртуальный канал, который представляет собой единственный маршрут, соединяющий эти конечные узлы.

Виртуальный канал может быть динамическим или постоянным.

Динамический виртуальный канал устанавливается при передаче в сеть специального пакета - запроса на установление соединения. Этот пакет проходит через коммутаторы и "прокладывает" виртуальный канал. Это означает, что коммутаторы запоминают маршрут для данного соединения и при поступлении последующих пакетов данного соединения отправляют их всегда по проложенному маршруту.

Постоянные виртуальные каналы создаются администраторами сети путем ручной настройки коммутаторов.

Более высокая эффективность сетей с коммутацией пакетов по сравнению с сетями с коммутацией каналов была доказана как экспериментально, так и с помощью имитационного моделирования.

Под ***коммутацией сообщений*** понимается передача единого блока данных между транзитными компьютерами сети с временной буферизацией этого блока на диске каждого компьютера.

Сообщение в отличие от пакета имеет произвольную длину, которая

определяется не технологическими соображениями, а содержанием информации, составляющей сообщение (текстовый документ, файл с кодом программы, электронное письмо).

Транзитные компьютеры могут соединяться между собой как сетью с коммутацией пакетов, так и сетью с коммутацией каналов. Сообщение хранится в транзитном компьютере на диске, причем время хранения может быть достаточно большим, если компьютер загружен другими работами или сеть временно перегружена.

По такой схеме обычно передаются сообщения, не требующие немедленного ответа, чаще всего сообщения электронной почты. Режим передачи с промежуточным хранением на диске называется ***режимом «хранение-и-передача» (store-and-forward)***.

Техника коммутации сообщений появилась в компьютерных сетях раньше техники коммутации пакетов, но потом была вытеснена последней, как более эффективной по критерию пропускной способности сети.

Метод уплотнения (мультиплексирования) сигнала

По методу уплотнения сигналов сети можно разделить на две большие категории:

- с временным уплотнением;
- с частотным уплотнением.

Эти два метода основаны на разных принципах использования полосы частот системы.

- ***Временное уплотнение***. В сетях с временным уплотнением (или с передачей сигнала без модуляции) в любой конкретный момент времени передачу данных через сеть ведет одно устройство, занимая всю полосу частот системы. Такой метод обеспечивает очень высокую скорость передачи в битах в секунду. Чтобы дать возможность всем абонентам общаться в сети, длительность каждой передачи должна ограничиваться заданным интервалом времени. К каждому блоку данных присоединяется адрес того узла, на который они должны пересылаться. Каждый узел постоянно контролирует адреса на шине и выделяет «свои» блоки. Общее число абонентов такой сети зависит от средней длины сообщений и их количества.

- ***Частотное уплотнение***. В сетях с частотным уплотнением (или широкополосных сетях) полоса частот системы разбита на ряд неперекрывающихся частотных диапазонов. Каждой паре взаимодействующих узлов выделяется один из этих поддиапазонов. Поэтому нет необходимости указывать адрес перед блоком данных и ограничивать длительность передачи. В любой момент времени обращаться к сети может много абонентов, однако на число одновременно взаимодействующих пар

накладывается естественное ограничение, зависящее от количества поддиапазонов. Поскольку многие изготовители используют в своих сетях системы кабельного телевидения, это позволяет организовать много сотен каналов обмена через один кабель.

Плезиохронная цифровая иерархия (Plesiochronous Digital Hierarchy; PDH) была разработана в лаборатории Bell Labs и предназначалась для передачи голосовой информации через каналы связи. Внедрение разработки должна была значительно повысить эффективность связи в городах При использовании этой технологии формируется иерархия из ***цифровых каналов (Digital Stream, DS)***, каждому из которых назначен уровень и номер. Цифровые потоки с меньшими номерами мультиплексируются в потоки с большими номерами с определенным сдвигом частоты.

Синхронная цифровая иерархия (Synchronous Digital Hierarchy, SDH) — это система передачи, определенная комитетом ITU. Первоначально SDH предлагали рассматривать эту технологию в качестве всеобщего всемирного стандарта и заменить ею существующую и уже устаревшую передающую инфраструктуру национальных и интернациональных сетей. Спецификация SDH описывает передачу кадров на физическом уровне эталонной модели OSI и может использоваться как один из транспортных механизмов для технологии ATM. В табл. 3.2 показаны уровни иерархии SDH.

Таблица 3.2. - Уровни иерархии SDH

Уровни	Скорость передачи (Мбит/с)
STM-0 или STM-1/3	51.84
STM-1	155.52
STM-4	622.08
STM-16	2488.32

Основное достоинство SDH, по сравнению со старой структурой PDH, за-ключается в прозрачности процесса мультиплексирования. Это означает, что базовый канал со скоростью 64 Кбит/с может быть выделен напрямую из уровней высшей иерархии SDH (в настоящее время 2.4 Гбит/с) и наоборот. По этой причине принцип функционирования сети SDH часто называется однофазным мультиплексированием, что отличается от сети PDH, в которой прямая локализация определенного коммуникационного канала (например, канала со скоростью 64 Кбит/с) в мультиплексированных каналах второго или третьего уровня иерархии (например, 140 Мбит/с) невозможна. Плезиохронный мультиплексор должен демультиплексировать весь поток информации для выделения нескольких компонентов сигналов, а затем

выполнить процедуру мультиплексирования заново. Мультиплексор SDH выделяет необходимые составляющие сигнала, не разбирая весь поток. При сравнении с технологией PDH, SDH позволяет избежать использования большого числа дорогих мультиплексирующих и демультиплексирующих устройств и разрабатывать более гибкую структуру сети.

Важными качествами SDH являются централизованное управление сетью с обеспечением полного мониторинга состояния каналов и узлов и автоматическое исключение неисправного оборудования с перемаршрутизацией каналов, что резко повышает надежность сети в целом.

ISDN (Integrated Services Digital Network), использует цифровые каналы связи в режиме коммутации каналов, а данные обрабатываются в цифровой форме. Первоначально ISDN задумывалась как сеть, способная интегрировать существующие телефонные сети с сетями передачи данных.

Адресация в сети строится по телефонному принципу. Номер ISDN состоит из 15 десятичных цифр и включает в себя код страны, код сети и код местной подсети. Код страны такой же, как в обычной телефонной сети. По коду сети выполняется переход в заданную сеть ISDN. Внутри подсети для адресации используется 35 десятичных цифр, что позволяет детально идентифицировать любое устройство.

Основным достоинством сетей ISDN является то, что они позволяют объединить в единое целое различные виды связи (передачу видео-, аудиоданных). Скорости передачи данных, реализуемые сетью: 64 Кбит/с, 128 Кбит/с, в более дорогих системах и до 2 Мбит/с, а в мощных сетях на широкополосных каналах связи до 155 Мбит/с.

Архитектура сети ISDN предусматривает несколько видов служб (рисунок 3.10):

- некоммутируемые средства (выделенные цифровые каналы);
- коммутируемая телефонная сеть общего пользования;
- сеть передачи данных с коммутацией каналов;
- сеть передачи данных с коммутацией пакетов;
- сеть передачи данных с трансляцией кадров (frame relay);
- средства контроля и управления работой сети.

Как видно из приведенного списка, транспортные службы сетей ISDN покрывают очень широкий спектр услуг, включая популярные услуги frame relay. Кроме того, большое внимание уделено средствам контроля сети, которые позволяют маршрутизировать вызовы для установления соединения с абонентом сети, а также осуществлять мониторинг и управление сетью. Управляемость сети обеспечивается интеллектуальностью коммутаторов и конечных узлов сети, поддерживающих стек протоколов, в том числе и

специальных протоколов управления.

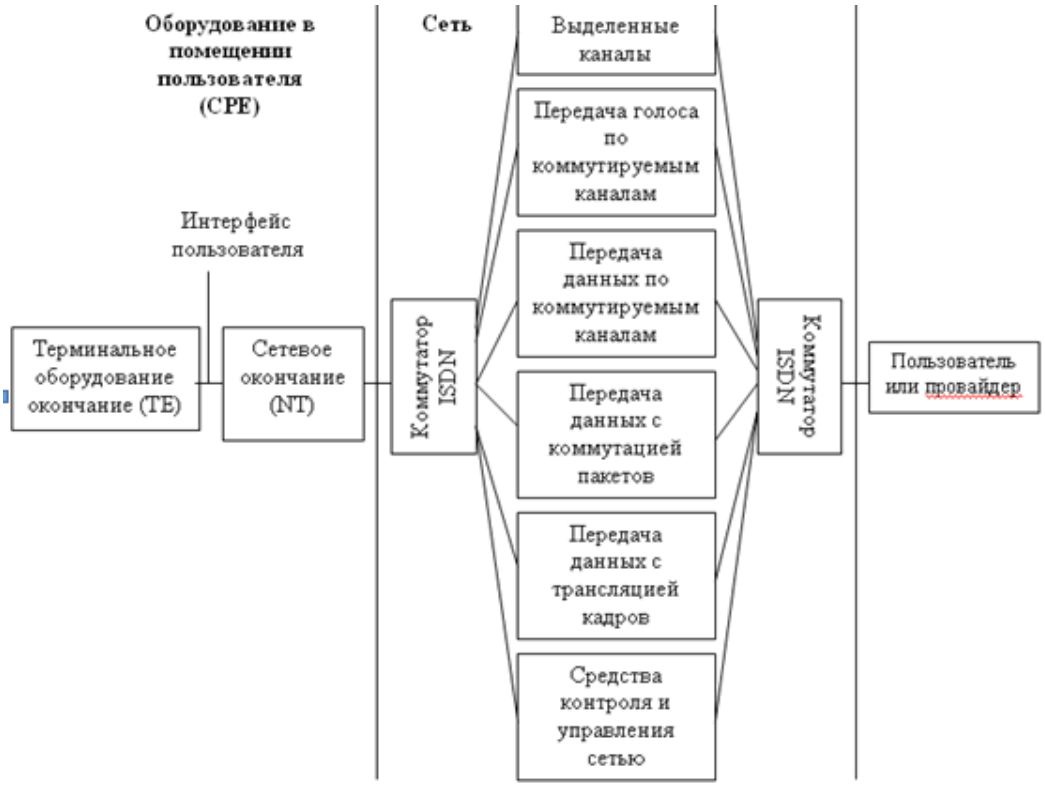


Рисунок 3.10 - Архитектура сети ISDN

Сети X.25 являются на сегодняшний день распространёнными сетями с коммутацией пакетов, используемыми для построения корпоративных сетей. Основная причина такой ситуации состоит в том, что долгое время сети X.25 были единственными доступными сетями с коммутацией пакетов коммерческого типа, в которых давались гарантии коэффициента готовности сети. Сети X.25 хорошо работают на ненадежных линиях благодаря протоколам с установлением соединения и коррекцией ошибок на двух уровнях — канальном и сетевом.

Стандарт X.25 «Интерфейс между оконечным оборудованием данных и аппаратурой передачи данных для терминалов, работающих в пакетном режиме в сетях передачи данных общего пользования» был разработан комитетом CCITT и пересматривался несколько раз. Стандарт наилучшим образом подходит для передачи трафика низкой интенсивности, характерного для терминалов, и в меньшей степени соответствует более высоким требованиям трафика локальных сетей. Как видно из названия, стандарт не описывает внутреннее устройство сети X.25, а только определяет пользовательский интерфейс с сетью. Взаимодействие двух сетей X.25 определяет стандарт X.75.

Сети X.25 базируются на следующих основополагающих принципах

организации, отличающих их от других:

• наличие в структуре сети специального устройства — ***PAD (Packet Assembler Disassembler),*** предназначенного для выполнения операции сборки нескольких низкоскоростных потоков байт от алфавитно-цифровых терминалов в пакеты, передаваемые по сети и направляемые компьютерам для обработки;

• наличие трехуровневого стека протоколов с использованием на канальном и сетевом уровнях протоколов с установлением соединения, управляющих потоками данных и исправляющих ошибки;

• ориентация на однородные стеки транспортных протоколов во всех узлах сети — сетевой уровень рассчитан на работу только с одним протоколом канального уровня и не может подобно протоколу IP объединять разнородные сети.

Дополнительными устройствами в сети X.25 являются также коммутаторы (центры коммутации пакетов), расположенные в различных географических областях и соединенные высокоскоростными каналами связи, обеспечивающими обмен данными между ними (рис. 3.11).

Рисунок 3.11 - Структура построения сети X.25

Сеть Frame Relay является сетью с коммутацией кадров или сетью с ретрансляцией кадров, ориентированной на использование цифровых линий связи. Первоначально технология Frame Relay была стандартизирована как служба в сетях ISDN со скоростью передачи данных до 2 Мбит/с. В дальнейшем эта технология получила самостоятельное развитие. Frame Relay поддерживает физический и канальный уровни OSI. Технология Frame Relay

использует для передачи данных технику виртуальных соединений (коммутируемых и постоянных).

Стек протоколов Frame Relay передает кадры при установленном виртуальном соединении по протоколам физического и канального уровней. В Frame Relay функции сетевого уровня перемещены на канальный уровень, поэтому необходимость в сетевом уровне отпала. На канальном уровне в Frame Relay выполняется мультиплексирование потока данных в кадры.

Каждый кадр канального уровня содержит заголовок, содержащий номер логического соединения, который используется для маршрутизации и коммутации трафика. Frame Relay - осуществляет мультиплексирование в одном канале связи нескольких потоков данных. Кадры при передаче через коммутатор не подвергаются преобразованиям, поэтому сеть получила название ретрансляции кадров. Таким образом, сеть коммутирует кадры, а не пакеты. Скорость передачи данных до 44 Мбит/с, но без гарантии целостности данных и достоверности их доставки.

Frame Relay ориентирована на цифровые каналы передачи данных хорошего качества, поэтому в ней отсутствует проверка выполнения соединения между узлами и контроль достоверности данных на канальном уровне. Кадры передаются без преобразования и контроля как в коммутаторах локальных сетей. За счет этого сети Frame Relay обладают высокой производительностью. При обнаружениях ошибок в кадрах повторная передача кадров не выполняется, а искаженные кадры отбраковываются. Контроль достоверности данных осуществляется на более высоких уровнях модели OSI.

Сети Frame Relay широко используется в корпоративных и территориальных сетях в качестве:

1) каналов для обмена данными между удаленными локальными сетями (в корпоративных сетях);

2) каналов для обмена данными между локальными и территориальными (глобальными) сетями.

Технология Frame Relay (FR) в основном используется для маршрутизации протоколов локальных сетей через общие (публичные) коммуникационные сети. Frame Relay обеспечивает передачу данных с коммутацией пакетов через интерфейс между оконечными устройствами пользователя DTE (маршрутизаторами, мостами, ПК) и оконечным оборудованием канала передачи данных DCE.

Коммутаторы Frame Relay используют технологию сквозной коммутации, т.е. кадры передаются с коммутатора на коммутатор сразу после прочтения адреса назначения, что обеспечивает высокую скорость передачи

данных. В сетях Frame Relay применяются высококачественные каналы передачи, поэтому возможна передача трафика чувствительного к задержкам (голосовых и мультимедийных данных). В магистральных каналах сети Frame Relay используются волоконно-оптические кабели, а в каналах доступа может применяться высококачественная витая пара.

Рисунок 3.12 - Структурная схема сети Frame Relay

На рисунке изображены основные элементы:

• ***DTE (data terminal equipment)*** – аппаратура передачи данных (маршрутизаторы, мосты, ПК).

• ***DCE (data circuit-terminating equipment)*** – оконечное оборудование канала передачи данных (телекоммуникационное оборудование, обеспечивающее доступ к сети).

Технология ***асинхронного режима передачи (Asynchronous Transfer Mode, ATM)*** - технология передачи данных является одной перспективных технологий построения высокоскоростных сетей (от локальных до глобальных). ATM - это коммуникационная технология, объединяющая принципы коммутации пакетов и каналов для передачи информации различного типа.

Технология ATM разрабатывалась для передачи всех видов трафика в локальных и глобальных сетях, т.е. передачи разнородного трафика (цифровых, голосовых и мультимедийных данных) по одним и тем же системам и линиям связи. Скорость передачи данных в магистралях ATM составляет 155 Мбит/с - 2200 Мбит/с.

ATM поддерживает физический и канальный уровни OSI. Технология ATM использует для передачи данных технику виртуальных соединений (коммутируемых и постоянных).

В технологии ATM информация передается в ячейках (cell) фиксированного размера в 53 байта, из них 48 байт предназначены для данных, а 5 байт - для служебной информации (для заголовка ячейки ATM). Ячейки не содержат адресной информации и контрольной суммы данных, что ускоряет их обработку и коммутацию.

20-байтовыми адресами приемник и передатчик обмениваются только в

момент установления виртуального соединения. Основная функция заголовка сводится к идентификации виртуального соединения. В процессе передачи информации ячейки пересылаются между узлами через сеть коммутаторов, соединенных между собой цифровыми линиями связи. В отличие от маршрутизаторов коммутаторы ATM выполняют свои функции аппаратно, что ускоряет чтение идентификатора в заголовке ячейки, после чего коммутатор переправляет ее из одного порта в другой.

Малый размер ячеек обеспечивает передачу трафика, чувствительного к задержкам. Фиксированный формат ячейки упрощает ее обработку коммуникационным оборудованием, которое аппаратно реализует функции коммутации ячеек.

Именно, сочетание фиксированного размера ячеек для передачи данных и реализация протоколов ATM в аппаратном обеспечении дает этой технологии возможность передавать все типы трафика по одним и тем же системам и линиям связи.

Телекоммуникационная сеть, использующая технологию ATM, состоит из набора коммутаторов, связанных между собой. Коммутаторы ATM поддерживают два вида интерфейсов: ***UNI (UNI - user-network interface)*** и ***NNI (NNI - network-network interface)***. Пользовательский интерфейс UNI (пользователь - сеть) используется для подключения к коммутатору конечных систем. Межсетевой интерфейс NNI (сеть - сеть) используется для соединений между коммутаторами.

ATM маршрутизатор состоит (рис 3.13):

- из маршрутизатора виртуальных путей;
- из маршрутизатор виртуальных каналов.

ATM маршрутизатор анализирует значения идентификаторов виртуального пути и виртуального канала ячейки, которая поступает на его вход и направляет ячейку на один из его выходных портов. Номер выходного порта определяется динамически создаваемой таблицей коммутации.

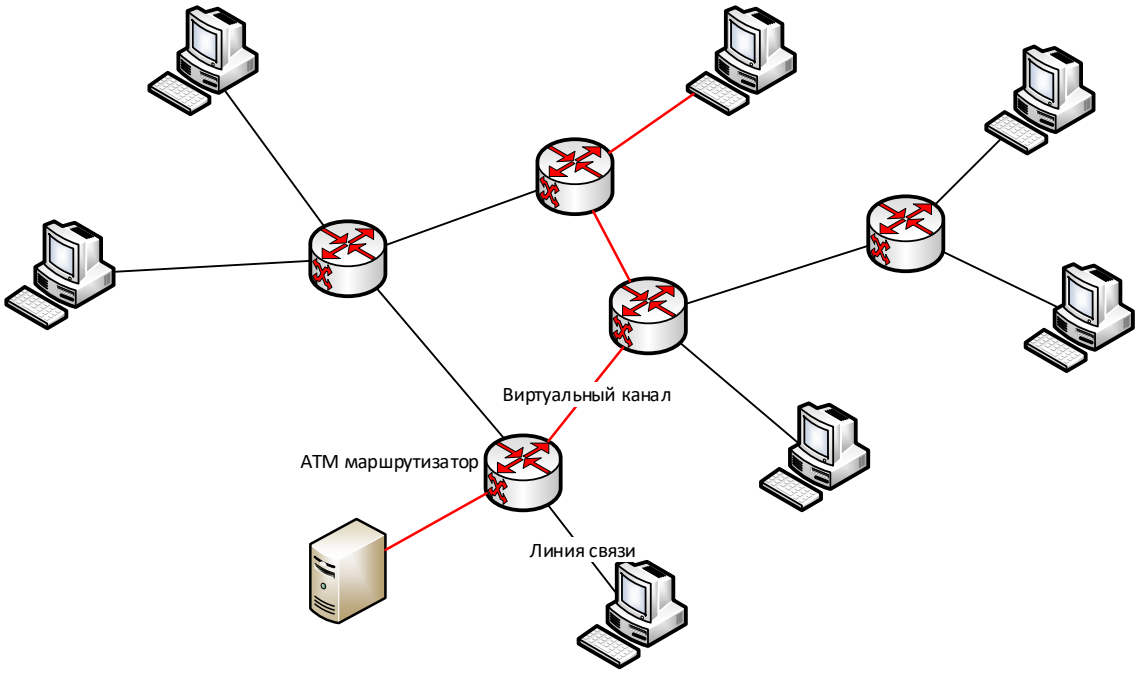


Рисунок 3.13 - Структурная схема сети Frame Relay

Для передачи данных в сети АТМ формируется виртуальное соединение. Виртуальное соединениеопределяется сочетанием идентификатора виртуального пути и идентификатора виртуального канала. Идентификатор позволяет маршрутизировать ячейку для доставки в путь назначения, т.е. коммутация ячеек происходит на основе идентификатора виртуального пути и идентификатора виртуального канала, определяющих виртуальное соединение. Несколько виртуальных путей составляют виртуальный канал.

Виртуальный канал является соединением, установленным между двумя конечными узлами на время их взаимодействия, а виртуальный путь – это путь между двумя коммутаторами.

При создании виртуального канала, коммутаторы определяют, какой виртуальный путь использовать для достижения пункта назначения. По одному и тому же виртуальному пути может передаваться одновременно трафик множества виртуальных каналов.

3.4. Глобальная сеть интернет

Интерне́т (сокращённое от INTERconnected NETworks – объединённые сети) – глобальная всемирная телекоммуникационная сеть, обеспечивающая связь для пересылки сообщений электронной почты, передачи файлов, соединения с другими компьютерами и получения доступа к информации, существующей в самых различных формах.

В ***1969 году, 29 октября в 9 вечера***, между первыми узлами данной сети, находящимися друг от друга на расстоянии в 640 километров – в Калифорнийском университете Лос-Анджелеса и в Стэндфордском исследовательском институте – провели первый сеанс связи. Оператор Чарли Клайн пытался выполнить удалённое подключение к компьютеру, находящемуся в Стэнфорде. Успешную передачу каждого введённого символа его коллега Билл Дювалль подтверждал по телефону. Вначале удалось отправить всего три символа «LOG», после чего сеть перестала работать. Символы «LOG» должны были быть словом LOGON (команда входа в систему). В рабочее состояние систему вернули уже к половине одиннадцатого вечера и следующая попытка оказалась успешной. Эту дату и принято считать днём появления сети Интернет.

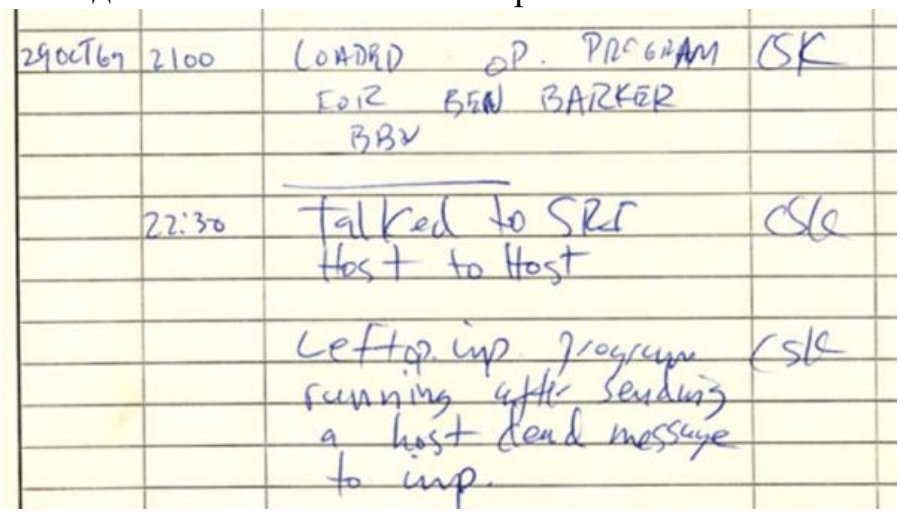


Рисунок 3.14 - ARPANET IMP журнал: первое сообщение, отправленное через ARPANET. 10:30 вечера, 29 октября 1969 года.

После первой успешной передачи данных в сети ARPANET следующим значимым этапом стала разработка в ***1971*** году первой программы для отправки электронной почты по сети.

К ***1973*** году в состав сети были включены первые зарубежные организации из Великобритании и Норвегии через трансатлантический телефонный кабель.

В 70-х годах прошлого века основным предназначением сети была пересылка электронной почты. В то же время появляются первые почтовые рассылки, различные доски объявлений и новостные группы. Однако во взаимодействии с другими сетями, построенными на других стандартах, были большие проблемы. Бурное развитие различных протоколов передачи данных, а так же их последующая стандартизация в 82-83 годах и переход на «общий», объединяющий протокол TCP/IP решили данную проблему. Этот переход

состоялся ***1 января 1983*** года. Именно в этом году сеть ARPANET закрепила за собой термин «Интернет».

Следующим этапом развития была разработка системы доменных имён (англ. Domain Name System, DNS), которая состоялась в ***1984*** году.

Так же в этом году появляется серьёзный конкурент сети ARPANET – межуниверситетская сеть NSFNet (англ. National Science Foundation Network). Эта сеть была объединением множества мелких сетей, имела пропускную способность гораздо бóльшую, чем у ARPANET, а так же высокую динамику подключения новых пользователей (около 10 тысяч машин в год). Название «Интернет» перешло к NSFNet.

В ***1988*** году был анонсирован протокол мгновенной передачи текстовых сообщений Internet Relay Chat (IRC), вследствии этого в Интернете стало возможным «живое» общение в чате в реальном времени.

В ***1989*** году знаменитый британский учёный Тим Бернерс-Ли предлагает концепцию Всемирной паутины. Он так же за два последующих года разрабатывает протокол HTTP, язык гипертекстовой разметки HTML и идентификаторы URI.

В ***1990*** году сеть ARPANET, проиграв в конкурентной борьбе NSFNet, прекращает своё существование. Так же в этом году состоялось первое подключение к сети Интернет по телефонной линии (Dialup access – «дозвон»).

1991 год ознаменовался общедоступностью Всемирной паутины в Интернете.

1993 год – появление знаменитого веб-браузера NCSA Mosaic. Быстрый рост популярности Интернета.

В ***1995*** году роль маршрутизации всего сетевого трафика Интернета возложили на себя сетевые провайдеры, а суперкомпьютеры NSFNet вернулись к роли исследовательской сети. В этом же году был образован Консорциум всемирной паутины (W3C), призванный упорядочить веб-стандарты.

С **1996** году Всемирная паутина (WWW) почти полностью подменяет собой понятие интернет, и обгоняет по трафику протокол пересылки файлов FTP.

1990-е годы произошло массовое объединение большинства существовавших сетей под флагом Интернет (хотя такие сети как Фидонет так и остались обособленными). Открытость технических стандартов во много способствовало быстрому росту сети. К ***1997*** году в Интернете насчитывалось около 10 млн. компьютеров и более 1 млн. доменных имён. Интернет – популярнейшее средство для обмена информацией.

Сейчас получить доступ в интернет можно через телефон, радио-каналы,

сотовую связь, спутники связи, кабельное телевидение, специальные оптико-волоконные линии и даже электропровода. А с 22 января 2010 года прямой доступ в Интернет появился и на Международной космической станции.

Непосредственно само построения глобальных компьютерных сетей основано на возможности передачи данных по различным телекоммуникационным системам между множеством региональных компьютерных сетей и компьютеров. Хранение информации на серверах, которые имеют свои адреса, обмен по высокоскоростным каналам связи информацией, необходимость наличия сетевых плат или сетевых адаптеров - вот основные принципы построения компьютерных сетей. Если рассматривать более подробно основные принципы построения компьютерных сетей, то стоит дать четкие определения основным понятиям. Большинство услуг Интернет строятся на основе принципа клиент-сервер.

Вся информация сети находиться на хранении на серверах. Подключение отдельных пользователей происходит через местных интернет-провайдеров, те в свою очередь подключены к крупным провайдерам, обладающим регионально расположенными узлами, которые образуют сеть транснациональных провайдеров, провайдеров первого уровня. Передача данных осуществляется в результате обеспечения сетевыми протоколами взаимодействия компьютеров разного типа, работающих под различными операционными системами.

Принципы построения глобальных компьютерных сетей – это набор правил, который позволяет создавать унифицированные и совместимые системы. Глобальные сети не ограничивают возможности индивидуальной потребности и не отрицают индивидуальность.

Главной особенностью таких сетей является наличие в каждом узле компьютера, который способен автономно выполнять некоторые задачи. Такие сети принято называть распределенными. Этим они отличаются, например, от централизованных терминальных сетей. Терминал по сложности оборудования может не уступать компьютеру, но его автономная работа не предусмотрена.

На текущий момент в следствии своего глобального распространения, интернет обеспечивают широкий спектр разнообразных информационных услуг, реализуемых различными службами:

- служба пересылки и приема сообщений (E-mail);
- служба гипертекстовой среды (WWW);
- служба передачи файлов (File Transfer Protocol - FTP);
- служба удаленного управления компьютером (Teletype Network - Telnet);

- служба имен доменов (Domain Name System);
- служба телеконференций (Users Network - Usenet) и чат-конференций (Интернет Relay Chat - IRC).

Программная индустрия для Web испытывает сейчас настоящий бум: сотни компаний - разработчиков программного обеспечения для Web создают новые технологии и инструментальные средства для навигации, работы в Сети и разработки пользовательских приложений. К их числу можно отнести:

- программы просмотра и навигации (браузеры);
- средства поиска и доставки информации (поисковые машины);
- программное обеспечение Интернет и Web-серверов, серверные приложения и расширения;
- средства администрирования в сетях;
- клиентские приложения и расширения (Web-сервисы);
- инструментальные средства разработки;
- средства обеспечения безопасности.

Инструментальные средства разработки Интернет-приложений разнообразны и включают:

- редакторы гипертекста и графические редакторы;
- средства разметки карт изображений и конверторы изображений;
- средства мультимедиа (аудио, анимация, видео);
- средства генерации виртуальной реальности;
- средства и языки программирования серверных и клиентских приложений и расширений.

Наиболее популярные услуги Интернета представлена в таблице 3.1.

Таблица 3.1 - Услуги Интернета

Услуги Интернета	
1	2
Всемирная паутина	Интернет-радио
Веб-форумы	Интернет-телевидение
Блоги	IP-телефония
Вики-проекты (и, в частности, Википедия)	Мессенджеры
Интернет-магазины	FTP-серверы
Интернет-аукционы	IRC (реализовано также как веб-чаты)
Социальные сети	Поисковые системы
Электронная почта и списки рассылки	Интернет-реклама

1	2
Группы новостей (в основном, Usenet)	Удалённые терминалы
Файлообменные сети	Удалённое управление
Электронные платёжные системы	web 2.0

Стоит остановиться на понятии ***всемирная паутина (World Wide Web)*** - распределенная система, предоставляющая доступ к связанным между собой документам, расположенным на различных компьютерах, подключенных к Интернету.

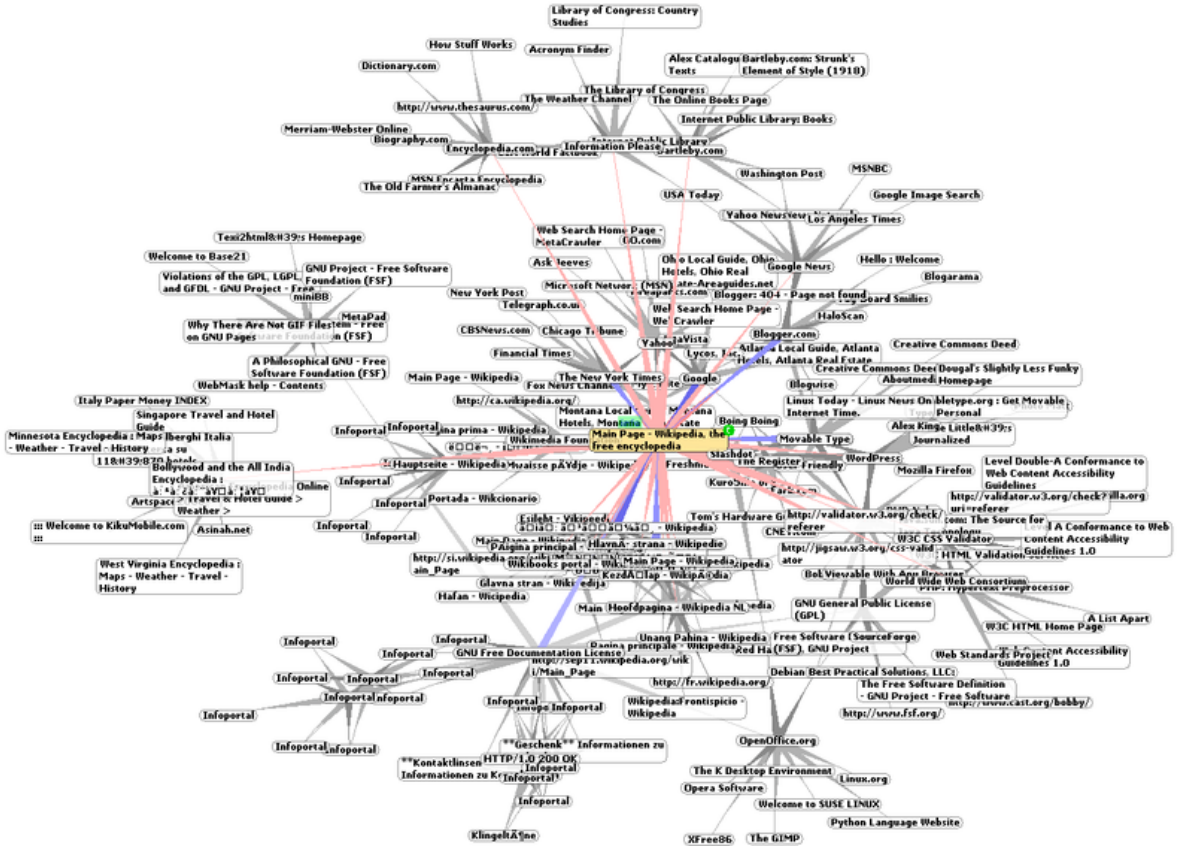


Рисунок 3.15 - Графическое изображение информации во Всемирной паутине

Всемирную паутину образуют более миллиона web-серверов. Большинство ресурсов всемирной паутины представляет собой гипертекст. Гипертекстовые документы, размещаемые во всемирной паутине, называются web-страницами. Несколько web-страниц, объединенных общей темой, дизайном, а также связанных между собой ссылками и обычно находящихся на одном и том же web-сервере, называются webсайтом. Для загрузки и просмотра web-страниц используются специальные программы браузеры.

Всемирная паутина вызвала настоящую революцию в информационных технологиях и бум в развитии Интернета. Часто, говоря об Интернете, имеют в виду именно Всемирную паутину, однако важно понимать, что это не одно и то же. Для обозначения Всемирной паутины также используют слово веб (web) и «WWW».

Чтобы найти в Интернете какой-либо документ, достаточно знать ссылку на него - так называемый ***универсальный указатель ресурса (URL - Uniform Resource Locator)***, который определяет местонахождение каждого файла, хранящегося на компьютере, подключенном к Интернету. Адрес URL является сетевым расширением понятия полного имени ресурса в операционной системе. В URL, кроме имени файла и директории, где он находится, указывается сетевое имя компьютера, на котором этот ресурс расположен, и протокол доступа к ресурсу, который можно использовать для обращения к нему. Система адресации URL и адресация почтовой службы имеют сходную структуру.

Рассмотрим структуру следующего URL: ***https://bntu.by/institutes/mido***

Первая часть ***http://***(***HyperText Transfer Protocol*** - протокол передачи гипертекста, по которому обеспечивается доставка документа с Web-сервера Web-браузеру) указывает программе просмотра (браузеру), что для доступа к ресурсу применяется данный сетевой протокол. В URL первым стоит указатель на тип доступа к запрашиваемому файлу (ресурсу), а затем его адрес. Вторая часть ***bntu.by*** указывает на доменное имя и адресует конкретный компьютер. Третья часть ***institutes/mido*** показывает программе-клиенту, где на данном компьютере-сервере искать ресурс.

Электронная почта (email, e-mail, от англ. electronic mail) - технология и предоставляемые ею услуги по пересылке и получению электронных сообщений (называемых «письма» или «электронные письма») по распределённой компьютерной сети. Основным отличием от прочих систем передачи сообщений является возможность отложенной доставки и развитая система взаимодействия между независимыми почтовыми серверами.

Программа, в которой пользователь формирует исходящие почтовые сообщения и просматривает входящие, является клиентом электронной почты (***почтовым клиентом***). Почтовый клиент должен быть настроен на определенный почтовый сервер. Иногда используют разные почтовые сервера: один для отправки исходящих сообщений (***SMTP-сервер, Sand Mail Transfer Protocol***), другой для получения входящей корреспонденции (***POP3***-сервер).

Общепринятым в мире протоколом обмена электронной почтой является SMTP (Simple mail transfer protocol, протокол передачи почты). В

общепринятой реализации он использует DNS для определения правил пересылки почты.

Адрес электронной почты имеет определенный формат: имя-***пользователя@[хосткомпьютер.]поддомен.домен_первого_уровня.*** Часть, выделенная в скобках [ ], является необязательной.

Адрес электронной почты не может включать символы кириллицы и пробелы, а также некоторые зарезервированные символы. Примеры e-mail адреса: Lina@mail.ru, In.Form@yandex.ru, travel_office@gmail.com.

Разделитель, стоящий в адресе после имени пользователя (@), который правильно именуется «коммерческая эт», или просто «эт».

Электронное письмо состоит из следующих частей: заголовков SMTP-протокола, полученных сервером, самого письма, которое в свою очередь состоит из следующих частей, разделённых пустой строкой: заголовков письма, тела письма (текст письма кодируется по стандарту MIME и не может быть прочитан человеком без использования декодера или почтового клиента).

Электронное письмо состоит из следующих частей: заголовков SMTP-протокола, полученных сервером, самого письма, которое в свою очередь состоит из следующих частей, разделённых пустой строкой: заголовков письма, тела письма (текст письма кодируется по стандарту MIME и не может быть прочитан человеком без использования декодера или почтового клиента).

Для организации передачи данных часто используется ***FTP (File Transfer Protocol)*** — стандартный протокол, предназначенный для передачи файлов по TCP-сетям.

Протокол построен на архитектуре "клиент-сервер" и использует разные сетевые соединения для передачи команд и данных между клиентом и сервером. Пользователи FTP могут пройти аутентификацию, передавая логин и пароль открытым текстом, или же, если это разрешено на сервере, они могут подключиться анонимно. Можно использовать протокол SSH для безопасной передачи, скрывающей (шифрующей) логин и пароль, а также шифрующей содержимое.

Особенность протокола FTP в том, что он использует множественное (как минимум - двойное) подключение. При этом один канал является управляющим, через который поступают команды серверу и возвращаются его ответы (обычно через TCP-порт 21), а через остальные происходит собственно передача данных, по одному каналу на каждую передачу. Поэтому в рамках одной сессии по протоколу FTP можно передавать одновременно несколько файлов, причём в обоих направлениях. Для каждого канала данных

открывается свой TCP порт, номер которого выбирается либо сервером, либо клиентом, в зависимости от режима передачи.

Передача данных может осуществляться в любом из трёх режимов:

1. ***Поточный режим*** - данные посылаются в виде непрерывного потока, освобождая FTP от выполнения какой бы то ни было обработки. Вместо этого, вся обработка выполняется TCP. Индикатор конца файла не нужен, за исключением разделения данных на записи.

2. ***Блочный режим*** - FTP разбивает данные на несколько блоков (блок заголовка, количество байт, поле данных) и затем передаёт их TCP.

3. ***Режим сжатия*** - данные сжимаются единым алгоритмом (обыкновенно, кодированием длин серий).

Существует несколько методов безопасной передачи файлов, в одно или другое время называемых "Безопасным FTP".

Явный ***FTPS*** - расширение стандарта FTP, позволяющее клиентам требовать того, чтобы FTPсессия была зашифрована. Это реализуется отправкой команды "AUTH TLS". Сервер обладает возможностью позволить или отклонить соединения, которые не запрашивают TLS. Это расширение протокола определено в спецификации RFC 4217. Неявный FTPS - устаревший стандарт для FTP, требующий использования SSL- или TLS-соединения. Этот стандарт должен был использовать отличные от обычного FTP порты.

SFTP, или «***SSH File Transfer Protocol***», не связан с FTP, за исключением того, что он тоже передаёт файлы и имеет аналогичный набор команд для пользователей. SFTP, или безопасный FTP, — это программа, использующая SSH (Secure Shell) для передачи файлов. В отличие от стандартного FTP он шифрует и команды, и данные, предохраняя пароли и конфиденциальную информацию от открытой передачи через сеть. По функциональности SFTP похож на FTP, но так как он использует другой протокол, клиенты стандартного FTP не могут связаться с SFTP-сервером и наоборот. FTP через SSH (не SFTP)

FTP через SSH относится к практике туннелирования обычной FTP-сессии через SSHсоединение. Поскольку FTP использует несколько TCP-соединений, туннелирование через SSH особенно затруднительно. Когда много SSH-клиентов пытаются установить туннель для канала управления (изначальное "клиент-сервер" соединение по порту 21), защищён будет только этот канал; при передаче данных программное обеспечение FTP на любом конце установит новые TCP-соединения (каналы данных), которые обойдут SSH-соединение и, таким образом, лишатся целостной защиты.

Таким образом Интернет, имеет два основных качественных значения:

•***глобальное сообщество*** произвольно объединяемых мировых сетей,

которые используются для свободного обмена данными, информацией и знаниями;

•***совокупность технологий***, которые реализуют обмен данными на основе использования семейства протоколов TCP/IP (Transmission Control Protocol/Internet Protocol), называемых Интернет-технологиями, и ***технологическая координация*** элементов интернета, в том числе управление системой доменных имен и распределение IP-адресов, а также выработкой и применением протоколов и стандартов.

Кроме того, необходимо понимать, что в контексте Интернета и ввиду его децентрализованного и открытого характера он должен являться общедоступным, и многообразие участников и форм регулирования является в настоящее время его неотъемлемым свойством.

РАЗДЕЛ 2. ПРАКТИЧЕСКИЙ

ЛАБОРАТОРНЫЕ РАБОТЫ

Лабораторная работа №1

Цель работы:

1. Изучить принципы работы сети Ethernet; кадр Ethernet; MAC-адрес.
2. Протокол IP. Адрес IP, сети и подсети, маска сети. Назначение прото¬колов ARP, RARP, ICMP. Маршрутизация (routing), шлюз (gateway), время жизни пакета TTL (на уровне понятий).
3. Ознакомится с протоколом NetBIOS. Назначение протокола NetBIOS и работа NetBIOS поверх TCP/IP.
4. Изучить сетевые команды оболочки Windows: arp, netstat, ipconfig, hostname, nslook

Теоретические сведения

Семейство протоколов ***Transmission Control Protocol/Intemet Protocol*** **(*TCP/IP*)** - это стандартный промышленный набор протоколов, разработанный для глобальных вычислительных сетей (***Wide Area Networks, WAN***).

Параметры конфигурации

Протокол TCP/IP использует IP-адрес, маску подсети и шлюз по умолчанию для соединения с узлами. Узлы TCP/IP, работающие в глобальной сети, требуют задания всех трех параметров в конфигурации. Каждая плата сетевого адаптера в компьютере, использующем TCP/IP, нуждается в этих параметрах.

IP-адрес

IP-адрес - это логический 32-разрядный адрес, однозначно определяю-щий узел TCP/IP. Каждый IP-адрес состоит из двух частей: идентификатора сети и идентификатора узла. Первый служит для обозначения всех узлов в одной физической сети. Второй обозначает конкретный узел сети. Каждому компьютеру, использующему TCP/IP, требуется уникальный IP-адрес.

Маска подсети

Маска подсети выделяет часть IP-адреса и позволяет TCP/IP отличить идентификатор сети от идентификатора узла. Пытаясь связаться, узлы TCP/IP используют маску подсети (например, 255.255.255.0), чтобы определить, находится узел-получатель в локальной или удаленной сети.

Шлюз по умолчанию

Для того чтобы установить соединение с узлом из другой сети,

необходимо сконфигурировать IP-адрес шлюза по умолчанию. TCP/IP посылает пакеты, предназначенные для удаленных сетей, на шлюз по умолчанию, но только в том случае, если на локальном узле не сконфигурирован другой маршрут к сети получателя. Если не сконфигурирован шлюз по умолчанию, то связь может быть ограничена локальной сетью.

Тестирование TCP/IP при помощи утилит Ipconfig и Ping

После того как установлен протокол TCP/IP, полезно проверить и протестировать конфигурацию и все соединения с другими узлами TCP/IP и сетями. Это можно сделать утилитами Ipconfig и Ping.

Утилита Ipconfig

Можно использовать утилиту Ipconfig для проверки параметров конфигурации узла, включая IP-адрес, маску подсети и шлюз по умолчанию. Это полезно при выяснении, успешно ли прошла инициализация TCP/IP и не дублируется ли IP-адрес, указанный в конфигурации.

Синтаксис команды:

ipconfig

Если протокол инициализировался успешно с заданной конфигурацией, то на экране отобразятся IP-адрес, маска подсети и шлюз по умолчанию. Если адрес, заданный в конфигурации уже используется другим узлом в сети на том же сегменте, то отобразится заданный IP-адрес, но маска подсети будет равна

Утилита Ping

После проверки конфигурации утилитой Ipconfig можно запустить утилиту Ping (***Packet InterNet Groper***) для тестирования соединений. Это диагностическое средство тестирует конфигурации TCP/IP и позволяет определить неисправности соединения. Утилита Ping использует пакеты эхо-запроса (***echo request***) и эхо-ответа (***echo reply***) протокола ICMP (***Internet Control Message Protocol***) для проверки доступности и работоспособности определенного узла TCP/IP.

Синтаксис команды:

ping IP_адрес

Если проверка прошла успешно, то отобразится сообщение типа:

Pinging IP_адрес with 32 bytes of data:
Reply from IP_адрес: bytes = x time<10ms TTL= x
Reply from IP_адрес: bytes = x time<10ms TTL= x
Reply from IP_адрес: bytes= x time<10ms TTL= x
Reply from IP_адрес: bytes= x time<10ms TTL= x

Общие сведения об именах NetBIOS

Имя NetBIOS назначается компьютеру в сети Microsoft Windows.

Стандарт NetBIOS, разработанный для IBM фирмой Sytek Corporation в 1983 году, позволяет приложениям взаимодействовать по сети. Этот стандарт определяет интерфейс сеансового уровня и протокол передачи данных и управления сеансом.

Интерфейс NetBIOS - доступный пользовательским приложениям стандартный прикладной интерфейс (API) для выполнения сетевого ввода/вывода и отправки команд управления к ПО нижележащего протокола. Прикладная программа, использующая интерфейс NetBIOS для сетевого взаимодействия, может работать с любым протоколом, поддерживающим интерфейс NetBIOS.

Стандартом NetBIOS определяется также протокол, действующий на сеансовом/транспортном уровне. Он реализуется в ПО нижележащего протокола, например NBFP (NetBEUI) или NetBT, где представлен весь набор команд сетевого уровня.

NetBIOS поддерживает следующие команды и функции:

- регистрацию и проверку сетевых имен;
- запуск и завершение сеанса;
- надежную передачу данных сеанса, ориентированного на соединение;
- ненадежную передачу датаграмм (datagram) без установки соединения;
- возможность мониторинга и управления протоколом (драйвером) и адаптером.

Команда Arp

Команда arp используется для просмотра, добавления или удаления записей в таблицах трансляции адресов IP в физические адреса. Эти записи используются при работе протокола *Address Resolution Protocol* (*ARP*)

Синтаксис команды:

arp -a [inet_addr] [-N [if_addr]]
arp -d inet_addr [if_addr]
arp -s inet_addr ether_addr [if_addr]

Команда Netstat

Производит отображение активных подключений TCP портов, прослушиваемых компьютером, статистики протокола Ethernet, таблицы маршрутизации IP, статистики семейства протоколов IPv4 (для протоколов IP, ICMP, TCP и UDP) и семейства протоколов IPv6 (для протоколов IPv6, ICMPv6, TCP через IPv6 и UDP через IPv6). Запущенная без параметров команда netstat отображает подключения TCP.

Синтаксис команды:

netstat [-a] [-e] [-n] [-o] [-p протокол] [-r] [-s] [интервал]

Чтобы вывести все активные подключения, отсортированные по

возрастанию номера порта, необходимо набрать:

netstat -n

Команда nslookup

Предоставляет сведения, предназначенные для диагностики инфраструктуры DNS. Для использования этого средства необходимо быть знакомым с принципами работы системы DNS. Средство командной строки nslookup доступно, только если установлен протокол TCP/IP

Синтаксис команды:

nslookup [-подкоманда ...] [{искомый_компьютер| [-сервер]}]

Команда Hostname

Утилита командной строки hostname выводит имя системы, на котором была запущена эта команда.

У данной команды нет параметров.

Команда Traceroute

Определяет путь до точки назначения с помощью посылки в точку назначения эхо-сообщений протокола ***Internet Control Message Protocol*** (***ICMP***) с постоянным увеличением значений срока жизни (***Time to Live, TTL***). Выведенный путь - это список ближайших адресов маршрутизаторов, находящихся на пути между узлом источника и точкой назначения. Ближний адрес представляют со бой адрес маршрутизатора, который является ближайшим к узлу отправителя на пути. Например, чтобы вывести трассу маршрута к http://www.microsoft.com, нужно выполнить команду:

tracert www.microsoft.com

Запущенная без параметров, команда tracert выводит справку.

Команда Route

Эта команда нужна для редактирования или просмотра таблицы маршрутов IP из командной строки. Параметр /? выводит все доступные параметры команды.

Задания

Программы пишутся на любом языке программирования, но должны использоваться только функции Windows API.

Написать программу, реализующую следующие функции:

1. Отображение MAC-адреса компьютера (можно воспользоваться функцией netbios).
2. Отображение всех рабочих групп, компьютеров в сети и их ресурсов (папок, открытых для общего доступа, принтеров). Воспользоваться функциями WNetXXX.

Пример программы получения MAC-адреса компьютера

*typedef struct _ASTAT_*

```
{ '
ADAPTER_STATUS adapt;
NAME_BUFFER NameBuff [30];
}ASTAT, * PASTAT;
ASTAT Adapter;
// Функция получения MAC адреса.
// На вход получает указатель на буфер, куда записывается строковое
// представление полученного MAC адреса.
BOOL GetMacAddress(char *buffer)
{
NCB neb;
UCHAR uRetCode;
char NetName[50];
memset( &ncb, 0, sizeof(ncb));
ncb.ncb_command = NCBRESET;
ncb.ncb_lana_num = 0;
uRetCode = Netbios( &ncb );
memset( &ncb, 0, sizeof(ncb));
ncb.ncb_command = NCBASTAT;
ncb.ncblananum = 0;

strcpy( (char *) ncb.ncbcallname, "*    ");
ncb.ncb_buffer = (unsigned char *) &Adapter;
ncb.ncb_length = sizeof( Adapter);

uRetCode = Netbios( &ncb );
if ( uRetCode == 0 )
{
sprintf(buffer, "%02X-%02X-%02X-%02X-%02X-%02X\n",
  Adapter.adapt.adapter_address [0],
  Adapter.adapt.adapter_address [1],
  Adapter.adapt.adapter_address [2],
  Adapter.adapt.adapter_address [3],
  Adapter.adapt.adapter_address [4],
  Adapter.adapt.adapter_address [5]);
return TRUE;
}
return FALSE;
}
```

Пример программы перечисления компьютеров в локальной сети

```
BOOL WIN API EnumerateFunc(HWND hwnd, HDC hdc, LPNETRESOURCE lpnr)
{
DWORD dwResult, dwResultEnum;
HANDLE hEnum;
DWORD cbBuffer = 16384;
DWORD cEntries = -1;    // Искать все объекты
LPNETRE SOURCE lpnrLocal;
DWORD i;

//Вызов функции WNetOpenEnum для начала перечисления компьютеров.
dwResult = WNetOpenEnum(RESOURCE_GLOBALNET, // все сетевые ресурсы
        RESOURCETYPEANY, // все типы ресурсов
        0,      // перечислить все ресурсы
        lpnrLocal, // равно NULL при первом вызове функции
        &hEnum); // дескриптор ресурса
if (dwResult! = NO ERROR)
{
// Обработка ошибок.
NetErrorHandler(hwnd, dwResult, (LPSTR)"WNetOpenEnum");
return FALSE;
}
 // Вызвов функции GlobalAlloc для выделения ресурсов.
lpnrLocal = (LPNETRESOURCE) GlobalAlloc(GPTR, cbBuffer);
if (lpnrLocal == NULL)
        return FALSE;
do
{
// Инициализируем буфер.
ZeroMemory(lpnrLocal, cbBuffer);
// Вызов функции WNetEnumResource для продолжения перечисления
// доступных ресурсов сети.
dwResultEnum = WNetEnumResource(hEnum,
        &cEntries, // Определено выше как -1
        lpnrLocal,
        &cbBuffer); // размер буфера
// Если вызов был успешен, то структуры обрабатываются циклом,
```

```
if (dwResultEnum == NO ERROR)
{
  for(i = 0; i < cEntries; i++)
{
// Вызов определенной в приложении функции для отображения
// содержимого структур NETRESOURCE.
DisplayStruct(hdc, &lpnrLocal[i]);
// Если структура NETRESOURCE является контейнером, то
// функции \EnumerateFunc вызывается рекурсивно.
if(RESOURCEUSAGE_CONTAINER == (lpnrLocal [i].dwUsage
          & RESOURCEUSAGE CONTAINER))
if(!EnumerateFunc(hwnd, hdc, &lpnrLocal[i]))
  TextOut(hdc, 10, 10, "EnumerateFunc returned FALSE.", 29);
}
}
// Обработка ошибок.
else if (dwResultEnum != ERROR_NO_MORE_ITEMS)
{
NetErrorHandler(hwnd, dwResultEnum, (LPSTR)"WNetEnumResource");
break;
}
}
while(dwResultEnum != ERROR_NO_MORE_ITEMS);

Вызов функции GlobalFree для очистки ресурсов.
GlobalFree((HGLOBAL)lpnrLocal);

// Вызов WNetCloseEnum для остановки перечисления.
dwResult = WNetCloseEnum(hEnum);

if(dwResult !=NO_ERROR)
{
// Обработка ошибок.
NetErrorHandler(hwnd, dwResult, (LPSTR)"WNetCloseEnum");
return FALSE;
}
return TRUE;
}
```

Лабораторная работа №2

Цель работы: изучить архитектуру прикладного интерфейса Windows Sockets (Winsock).

Теоретические сведения

Протокол TCP

Протокол TCP предоставляет надежную, ориентированную на соединение службу доставки.

Данные протокола TCP передаются сегментами, и соединение должно быть установлено до того, как узлы начнут обмениваться данными. TCP использует потоки, в которых данные представлены в виде последовательности байт.

TCP обеспечивает надежность, присваивая номера последовательности (sequence number) каждому передаваемому сегменту. Если сегмент разбивается на мелкие пакеты, то узел-получатель сможет узнать, все ли части получены. Для этого используются подтверждения. Для каждого отправленного сегмента узел-получатель должен вернуть отправителю подтверждение (acknowledgement, ACK) в течение определенного времени.

Если отправитель не получил ACK, то данные передаются повторно. Если сегмент поврежден, то узел-получатель отвергает его. Поскольку ACK в этом случае не посылается, отправитель передает сегмент еще раз.

Примечание: TCP описан в стандарте RFC 793

Порты

Приложения, использующие «сокеты», идентифицируют себя на компьютере посредством номера порта (port number). Например, FTP-сервер использует определенный TCP-порт, поэтому другие приложения могут связаться с ним.

Порты могут иметь любой номер от 0 до 65 536. Номера портов для приложений клиентов динамически назначаются операционной системой при обработке запроса на обслуживание. Существуют зарезервированные (или заранее известные, от англ, well-known) номера портов, которые закреплены за стандартными прикладными протоколами и службами. Их список можно узнать, просмотрев текстовый файл Windows\System32\Drivers\Etc\Services.

Номера зарезервированных портов расположены в интервале от 1 до 1024. Полный список зарезервированных номеров портов определен в стандарте RFC 1700.

«Сокеты»

«Сокет» (от англ, socket) во многом аналогичен дескриптору файла (file handle). Он обеспечивает конечную точку сетевого соединения. Приложение,

создавая «сокет», указывает три параметра: IP-адрес узла, тип обслуживания (протокол TCP для ориентированного на соединение обслуживания и UDP для не ориентированного) и порт, используемый приложением.

Протокол UDP

Протокол User Datagram Protocol (UDP) обеспечивает не ориентированную на соединение службу доставки датаграмм по принципу «максимального усилия». Это означает, что получение всей датаграммы или правильной последовательности отправленных пакетов не гарантируется.

Протокол UDP используется приложениями, не требующими подтверждения. Обычно такие приложения передают данные небольшого объема за один раз. Примеры служб и приложений, использующих UDP: сервис имен NetBIOS, сервис датаграмм NetBIOS и сервис SNMP.

Порты протокола UDP

Для использования протокола UDP приложение должно знать IP-адрес и номер порта получателя. Порт - место назначения при доставке сообщений - идентифицируется уникальным номером. Порт действует как мультиплексная очередь сообщений, то есть он может получать несколько сообщений одновременно. Важно отметить, что порты протокола UDP, перечисленные в таблице, отличаются от портов TCP несмотря на использование тех же значений номеров. Протокол UDP описан в стандарте RFC 768.

Сокеты в ОС Windows

В ОС Windows для работы с «сокетами» используется библиотека Winsock, которая представляют собой высокоуровневый унифицированный интерфейс взаимодействия с телекоммуникационными протоколами. Основное подспорье в изучении сокетов - Windows Sockets 2 SDK (от англ. Software Development Kit) - документация, набор заголовочных файлов и инструментарий разработчика.

Библиотека Winsock поддерживает два вида сокетов - синхронные (блокируемые) и асинхронные (неблокируемые). Синхронные сокеты задерживают управление на время выполнения операции, а асинхронные возвращают его немедленно, продолжая выполнение в фоновом режиме, и, закончив работу, уведомляют об этом вызывающий код.

Независимо от вида, сокеты делятся на два типа - потоковые и дейтаграммные. Потоковые сокеты работают с установкой соединения, обеспечивая надежную идентификацию обоих сторон и гарантируют целостность и успешность доставки данных. Дейтаграмные сокеты работают без установки соединения и не обеспечивают ни идентификации отправителя, ни контроля успешности доставки данных, зато они заметно быстрее потоковых.

Выбор того или иного типа сокетов определяется транспортным протоколом на котором работает сервер, - клиент не может по своему желанию установить с дейтаграммным сервером потоковое соединение.

Замечание: дейтаграммные сокеты опираются на протокол UDP, а потоковые на TCP.

Перед началом использования функций библиотеки Winsock ее необходимо подготовить к работе вызовом функции

int WSAStartup (WORD wVersionRequested, LPWSADATA lpWSA-Data)

передав в старшем байта слова wVersionRequested номер требуемой версии, а в младшем - номер подверсии.

Второй шаг - создание объекта "сокет". Это осуществляется функцией

SOCKET socket (int af, int type, int protocol)

Первый слева аргумент указывает на семейство используемых протоколов. Для интернет-приложений он должен иметь значение AFINET. Следующий аргумент задает тип создаваемого сокета - потоковый (SOCK_STREAM) или дейтаграммный (SOCK_DGRAM). Последний аргумент уточняет, какой транспортный протокол следует использовать. Нулевое значение соответствует выбору по умолчанию: TCP для потоковых сокетов и UDP для дейтаграммных. Если функция завершилась успешно, она возвращает дескриптор сокета, в противном случае значение INVALID_SOCKET.

Дальнейшие шаги зависит от того, является приложение сервером или клиентом. Ниже эти два случая будут описаны раздельно.

Клиент: шаг третий - для установки соединения с удаленным узлом потоковый сокет должен вызвать функцию

int connect (SOCKET s, const struct sockaddr FAR name, int namelen)*

Датаграмные сокеты работают без установки соединения, поэтому, обычно не обращаются к функции connect.

Первый слева аргумент - дескриптор сокета, возращенный функцией socket; второй - указатель на структуру "sockaddr", содержащую в себе адрес и порт удаленного узла с которым устанавливается соединение. Последний аргумент сообщает функции размер структуры sockaddr.

После вызова connect система предпринимает попытку установить соединение с указанным узлом. Если по каким-то причинам это сделать не удастся (адрес задан неправильно, узел не существует или "висит", компьютер находится не в сети), функция возвратит ненулевое значение.

Сервер: шаг третий - прежде, чем сервер сможет использовать сокет, он должен связать его с локальным адресом. Локальный, как, впрочем, и любой другой адрес Интернета, состоит из IP-адреса узла и номера порта. Если сервер

имеет несколько IP адресов, то сокет может быть связан как со всеми сразу (для этого вместо IP-адреса следует указать константу INADDR_ANY равную нулю), так и с каким-то конкретным одним.

Связывание осуществляется вызовом функции

int bind (SOCKET s, const struct sockaddr FAR name, int namelen)*

Первым слева аргументом передается дескриптор сокета, возращенный функций socket, за ним следуют указатель на структуру sockaddr и ее длина.

Строго говоря, клиент также должен связывать сокет с локальным адресом перед его использованием, однако, за него это делает функция connect, ассоциируя сокет с одним из портов, наугад выбранных из диапазона 1024-5000. Сервер же должен "садиться" на заранее определенный порт, например, 21 для FTP, 23 для telnet, 25 для SMTP, 80 для WEB, 110 для POP3 и т.д. Поэтому ему приходится осуществлять связывание "вручную".

При успешном выполнении функция возвращает нулевое значение и ненулевое в противном случае.

Сервер: шаг четвертый - выполнив связывание, потоковый сервер переходит в режим ожидания подключений, вызывая функцию

int listen (SOCKET s, int backlog),

где s - дескриптор сокета, a backlog - максимально допустимый размер очереди сообщений.

Размер очереди ограничивает количество одновременно обрабатываемых соединений, поэтому, к его выбору следует подходить "с умом". Если очередь полностью заполнена, очередной клиент при попытке установить соединение получит отказ (TCP пакет с установленным флагом RST). В то же время максимально разумное количество подключений определяются производительностью сервера, объемом оперативной памяти и т.д.

Датаграммные серверы не вызывают функцию listen, т.к. работают без установки соединения и сразу же после выполнения связывания могут вызывать recvfrom для чтения входящих сообщений, минуя четвертый и пятый шаги.

Сервер: шаг пятый - извлечение запросов на соединение из очереди осуществляется функцией

SOCKET accept (SOCKET s, struct sockaddr FAR addr, int FAR* addrlen),*

которая автоматически создает новый сокет, выполняет связывание и возвращает его дескриптор, а в структуру sockaddr заносит сведения о подключившемся клиенте (IP-адрес и порт). Если в момент вызова accept очередь пуста, функция не возвращает управление до тех пор, пока с сервером не будет установлено хотя бы одно соединение. В случае возникновения

ошибки функция возвращает отрицательное значение.

Для параллельной работы с несколькими клиентами следует сразу же после извлечения запроса из очереди порождать новый поток (процесс), передавая ему дескриптор созданного функцией accept сокета, затем вновь извлекать из очереди очередной запрос и т.д. В противном случае, пока не завершит работу один клиент, север не сможет обслуживать всех остальных.

Клиент и сервер: после того как соединение установлено, потоковые сокеты могут обмениваться с удаленным узлом данными, вызывая функции

*int send (SOCKET s, const char FAR * buf, int len,int flags)*

int recv (SOCKET s, char FAR buf, int len, int flags)*

для посылки и приема данных соответственно.

Функция send возвращает управление сразу же после ее выполнения независимо от того, получила ли принимающая сторона наши данные или нет. При успешном завершении функция возвращает количество передаваемых (не переданных!) данных - т.е. успешное завершение еще не свидетельствует об успешной доставке! В общем-то, протокол TCP (на который опираются потоковые сокеты) гарантирует успешную доставку данных получателю, но лишь при условии, что соединение не будет преждевременно разорвано. Если связь прервется до окончания пересылки, данные останутся не переданными, но вызывающий код не получит об этом никакого уведомления! А ошибка возвращается лишь в том случае, если соединение разорвано до вызова функции send.

Функция же recv возвращает управление только после того, как получит хотя бы один байт. Точнее говоря, она ожидает прихода целой дейтаграммы. Дейтаграмма - это совокупность одного или нескольких IP пакетов, посланных вызовом send. Упрощенно говоря, каждый вызов recv за один раз получает столько байтов, сколько их было послано функцией send. При этом подразумевается, что функции recv предоставлен буфер достаточных размеров, - в противном случае ее придется вызвать несколько раз. Однако, при всех последующих обращениях данные будут браться из локального буфера, а не приниматься из сети, т.к. TCP-провайдер не может получить "кусочек" дейтаграммы, а только ею всю целиком.

Дейтаграммный сокет так же может пользоваться функциями send и recv, если предварительно вызовет connect (см. "Клиент: шаг третий"), но у него есть и свои, "персональные", функции:

*int sendto (SOCKET s, const char FAR * buf, int len,int flags, const struct sockaddr FAR * to, int tolen)*

int recvfrom (SOCKET s, char FAR buf, int len, int flags, struct sockaddr FAR* from, int FAR* fromlen)*

Они очень похожи на send и recv, - разница лишь в том, что sendto и recvfrom требуют явного указания адреса узла принимаемого или передаваемого данные. Вызов recvfrom не требует предварительного задания адреса передающего узла - функция принимает все пакеты, приходящие на заданный UDP- порт со всех IP адресов и портов. Напротив, отвечать отправителю следует на тот же самый порт откуда пришло сообщение. Поскольку, функция recvfrom заносит IP-адрес и номер порта клиента после получения от него сообщения, программисту фактически ничего не нужно делать - только передать sendto тот же самый указатель на структуру sockaddr, который был ранее передан функции recvfrem, получившей сообщение от клиента.

Еще одна деталь - транспортный протокол UDP, на который опираются дейтаграммные сокеты, не гарантирует успешной доставки сообщений и эта задача ложиться на плечи самого разработчика. Решить ее можно, например, посылкой клиентом подтверждения об успешности получения данных. Правда, клиент тоже не может быть уверен, что подтверждение дойдет до сервера, а не потеряется где-нибудь в дороге. Подтверждать же получение подтверждения - бессмысленно, т. к. это рекурсивно неразрешимо. Лучше вообще не использовать дейтаграммные сокеты на ненадежных каналах.

Во всем остальном обе пары функций полностью идентичны.

Все четыре функции при возникновении ошибки возвращают значение *SOCKETERROR = -1.*

Шаг шестой, последний - для закрытия соединения и уничтожения сокета предназначена функция "int closesocket (SOCKET s)", которая в случае удачного завершения операции возвращает нулевое значение.

Перед выходом из программы, необходимо вызвать функцию "int WSACleanup (void)" для деинициализации библиотеки WINSOCK и освобождения используемых этим приложением ресурсов. Внимание: завершение процесса функцией ExitProcess автоматически не освобождает ресурсы сокетов.

Задания

Все программы выполняются на любом языке программирования, используются только функции Windows APE Для каждого варианта должно быть реализовано две версии программы: одна работающая посредством протокола TCP, вторая - UDP.

Написать программу, реализующую следующие функции:

Вариант 1: Программа представляет простейший «чат» (от англ, chat), позволяющий обмениваться текстовыми сообщениями между двумя компьютерами.

Вариант 2: Программа позволяет копировать бинарные файлы с одного компьютера на другой.

Вариант 3: Программа измеряет скорость передачи информации по протоколам TCP и ЕГОР, а так же количество потерянных (искаженных) пакетов. Трафик генерируется псевдослучайным образом (т.е. генерируется псевдослучайная последовательность данных, отсылается на другой компьютер и там сравнивается с эталоном).

Вариант 4: Программа удаленного рисования. На первом компьютере пользователь может рисовать кривые мышкой на «холсте». На втором на таком же холсте рисунок повторяется в реальном времени.

Пример программы реализации сервера DayTime

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

#define fatal(x) { perror(x); exit(l); }

main()
{
        int s, c, sz;
      struct sockaddrin ssa, csa;
      struct sockaddr *sp, *cp;
      char *tstr, node[128], service[16];
      time_t itime;

      sp = (struct sockaddr *)&ssa;
      cp = (struct sockaddr *)&csa;

      /* Создаём сокет */
      s = socket(AF_INET, SOCKSTREAM, 0);
      if (s == -1)
            fatal("Невозможно создать сокет");

       /* Занимаем порт 13 */
       ssa.sin_family = AFINET;
       ssa.sin_port = htons(13);
```

```
        ssa.sinaddr.s_addr = INADDR_ANY;

        if (bind(s, sp, sizeof(ssa)) == -1)
                fatal("Невозможно занять порт");

        /*
        *       Переводим сокет в режим ожидания запросов
        *       на установление соединения
        */
        if (listen(s, 0) == -1)
                fatal("Ошибка при выполнении listen");

        while (1) {
                /* Принимаем соединение */
                sz = sizeof(csa);
                if ((c = accept(s, cp, &sz)) == -1)
                        fatal("Ошибка при выполнении accept");
                /* Получаем строку, содержащую дату и время */
                itime = time(NULL);
                tstr = ctime(&itime);

                /* Выводим время поступления запроса */
                printf("%s\tполучен запрос от ", tstr);
                /* Выводим информацию о клиенте */
                if (getnameinfo(cp, sz, node,
                                128, service, 16, 0) == 0)
                        printf("%s:%s\n", node, service);
                else
                        printf("%s:%d\n", inet ntoa(csa.sin addr),
                                ntohs(csa.sin_port));

                /* Отправляем дату и время клиенту */
                send(c, tstr, 25, 0);

                /* Закрываем соединение */
                close(c);
        }
}
```

Лабораторная работа №3

Цель работы: изучить протоколы Echo, Time, DayTime, Wliols, Finger, RLogin, Telnet.

Теоретические сведения

Протокол Echo

Эхо-сервис весьма полезен для отладки и выполнения измерений. Этот сервис просто возвращает отправителю любые данные, полученные от него. Полное описание протокола можно найти в стандарте RFC 862.

Службы Echo на базе протокола TCP.

Один из вариантов эхо-сервиса определен как основанный на организации соединений приложение TCP. Эхо-сервер прослушивает соединения TCP на порту 7. После организации соединения все полученные через это соединение данные возвращаются отправителю. Процесс возврата полученных данных отправителю продолжается до тех пор, пока инициатор соединения не разорвет это соединение.

Службы Echo на базе протокола UDP.

Другой вариант эхо-сервиса не использует прямых соединений и основан на передаче дейтаграмм UDP. Эхо-сервер прослушивает порт 7 (ЕГОР) и возвращает отправителю все принятые через этот порт дейтаграммы.

Time

Данный протокол предназначен для синхронизации времени. В сети работают time-серверы, у которых можно запросить точное время. Следует заметить, что в настоящее время для синхронизации времени в глобальных сетях используется более сложный протокол - ***NTP*** - ***Network Time Protocol***. В ответ на запрос клиента, сервер возвращает время в секундах (32х битное двоичное число), прошедшее с 00:00:00 1 января 1900 года.

Этот протокол может использовать в качестве транспортной службы как UDP-протокол, так и TCP-протокол. Стандартный порт протокола - 37.

Если в качестве транспортной службы используется TCP, взаимодействие осуществляется так:

SERVER: прослушивает 37 порт, ожидая соединений
CLIENT: запрашивает соединение с портом 37 сервера
SERVER: посылает время в виде двоичного 32х битного числа
CLIENT: получает время
SERVER: закрывает соединение
CLIENT: закрывает соединение

Если сервер по каким-либо причинам не может определить время на своей стороне, он отказывается от соединения, не посылая ничего.

Если в качестве транспортной службы используется UDP, взаимодействие осуществляется так:

SERVER: прослушивает 37 порт, ожидая соединений

CLIENT: посылает серверу пустой UDP-пакет, номер порта = 37

SERVER: получает пустой UDP-пакет

SERVER: посылает UDP-пакет, содержащий время в виде двоичного 32х битного числа

CLIENT: получает UDP-пакет, содержащий время

Если сервер по каким-либо причинам не может определить время на своей стороне, он отбрасывает полученный пустой ЕГОР-пакет и не посылает ничего в ответ.

DayTime

Протокол DayTime определен в документе RFC867. Протокол может использовать в качестве транспортного протокола UDP и TCP. В случае использования ЕГОР сервер занимает 13-й порт и ожидает поступления датаграмм. После получения датаграммы он отправляет по обратному адресу пакет, содержащий текущие дату и время в виде текстовой строки. Дополнительно, сервер выводит информацию о поступившем запросе на стандартный вывод. В случае использования TCP, как и в UDP, сервер занимает порт 13 и ожидает поступления запросов на установление соединения. После установления соединения, сервер отправляет клиенту строку, содержащую дату и время, и закрывает соединение.

Whols

Протокол Whois это информационный сервис. Несмотря на то, что любой узел может предоставить Whois сервис, наиболее широко используется InterNIC, rs.intemic.net. Этот сервер содержит информацию обо всех зарегистрированных DNS доменах и о большинстве системных администраторов, которые ответственны за системы, подключенные к Internet. (Еще один подобный сервер nic.ddn.mil содержит информацию о сети MTLNET.) К сожалению, не всегда предоставляется полная информация. RFC 954 [Harrenstein, Stahl, and Feinler 1985] документирует сервис Whois.

С точки зрения протокола, сервер Whois работает с заранее известным портом TCP 43. Он принимает от клиента запрос на соединение, после чего клиент отправляет на сервер запрос длиной в 1 строку. Сервер выдает информацию и закрывает соединение. Запросы и отклики передаются в формате NVT ASCII. Он практически идентичен серверу Finger, за исключением того, что запросы и отклики содержат разную информацию.

Широко используемый Unix клиент - программа Whois, однако можно использовать Telnet и ввести команды самостоятельно. Сначала отправляется

запрос, содержащий знак вопроса, на что возвращается более подробная информация о поддерживаемых запросах клиента.

Finger

Протокол Finger возвращает информацию об одном или нескольких пользователях на указанном хосте. Это приложение обычно используется, для того чтобы посмотреть, находится ли конкретный пользователь в настоящее время в системе, или чтобы получить имя какого-либо пользователя, чтобы послать ему почту. RFC 1288 [Zimmerman 1991] описывает этот протокол.

Многие узлы не запускают Finger сервер по двум причинам. Во-первых, ошибки в программировании в ранних версиях сервера были одной из точек входа "червяка" в Internet в 1988 году. Во-вторых, протокол Finger может предоставить подробную информацию о пользователях (имя входа в систему, телефонные номера, время последнего входа и так далее), а эту информацию большинство администраторов считают частной. Раздел 3 RFC 1288 детально описывает аспекты секретности, соответствующие сервису Finger.

Сервер Finger использует заранее известный порт 79. Клиент осуществляет активное открытие на этот порт и отправляет запрос длиной в 1 строку. Сервер обрабатывает запрос, посылает назад вывод и закрывает соединение. Запрос и отклик в формате NVT ASCII, почти так же как и в случае FTP и SMTP.

Обычно большинство пользователей Unix получают доступ к серверу Finger с использованием клиента finger, однако можно воспользоваться Telnet клиентом. Если запрос клиента состоит из пустой строки (которая в NVT ASCII передается как CR, за которой следует LF), это воспринимается как запрос на информацию о всех текущих пользователях.

RLogin

Rlogin появился в 4.2BSD и был предназначен для захода удаленным терминалом между Unix хостами. Поэтому Rlogin проще, чем Telnet, так как он не требует определения параметров, которые для одной и той же операционной системы известны заранее и для клиента, и для сервера. Через несколько лет Rlogin был перенесен на не-Unix системы.

RFC 1282 [Kantor 1991] содержит спецификацию протокола Rlogin. Однако, как и в случае с RFC посвященным RIP (Routing Information Protocol), он был написан после того, как Rlogin уже использовался в течении нескольких лет.

Rlogin использует одно TCP соединение между клиентом и сервером. После того как TCP соединение установлено, между клиентом и сервером осуществляется следующая последовательность действий.

Клиент отправляет серверу четыре строки: нулевой байт, имя

пользователя на хосте клиента, заканчивающееся нулевым байтом, имя пользователя на хосте сервера, заканчивающееся нулевым байтом, тип терминала пользователя, за которым следует слэш, затем следует скорость терминала, и все это заканчивается нулевым байтом. Необходимо отправить именно два имени пользователя, потому что пользователи не обязаны иметь одинаковые имена на разных хостах.

Тип терминала передается от клиента к серверу, потому что эта информация необходима для большинства полноэкранных приложений. Скорость терминала передается, потому что некоторые приложения работают по-разному в зависимости от скорости.

Сервер отвечает нулевым байтом.

У сервера есть опция, с помощью которой он просит ввести пароль. Это осуществляется как обычный обмен данными по Rlogin соединению - специальные протоколы не применяются. Сервер отправляет клиенту строку (которую клиент отображает на терминале), чаще всего эта строка выглядит как «Password:». Если клиент не вводит пароль в течение определенного времени (обычно 60 секунд), сервер закрывает соединение. Все что вводится в ответ на приглашение сервера ввести пароль, передается в виде открытого текста. Символы введенного пароля посылаются так, как они есть. Каждый, кто может прочитать пакеты в сети, может прочитать любой пароль.

Сервер обычно отправляет запрос клиенту, спрашивая размер окна терминала.

Клиент посылает за один раз серверу 1 байт, каждый байт сервер отражает эхо-откликом. Функционально все довольно просто: то, что вводит пользователь, отправляется на сервер, а то, что сервер отправляет клиенту, отображается на терминале.

Telnet

Telnet был разработан, для того чтобы работать между хостами работающими под управлениием любых операционных систем, а также с любыми терминалами. Его спецификация, приведенная в RFC 854 [Postel and Reynolds 1983a], определяет терминал, который может являться наиболее общим, и который называется виртуальным сетевым терминалом (NVT - network virtual terminal). NVT это воображаемое устройство, находящееся на обоих концах соединения, у клиента и сервера, с помощью которого устанавливается соответствие между их реальными терминалами. Таким образом, операционная система клиента должна определять соответствие между тем типом терминала, за которым работает пользователь, с NVT. В свою очередь, сервер должен устанавливать соответствие между NVT и теми типами терминалов, которые он (сервер) поддерживает.

NVT это символьное устройство с клавиатурой и принтером. Данные, введенные пользователем с клавиатуры, отправляются серверу, а данные, полученные от сервера, поступают на принтер. По умолчанию клиент отражает эхом на принтер все, что ввел пользователь, однако, ниже увидим что, существуют опции, которые позволяют изменить подобное поведение.

Задания

Все программы выполняются на любом языке программирования, возможно использование как функций Windows API, так и любых других библиотек работы с сокетами.

Написать программу, реализующую следующие функции клиента и сервера одного из протоколов:

Вариант 1: Протокол Echo.

Вариант 2: Протокол Time.

Вариант 3: Протокол DayTime.

Вариант 4: Протокол Wliols.

Вариант 5: Протокол Finger.

Вариант 6: Протокол RLogin.

Вариант 7: Протокол Telnet.

Программа должна производить полную обработку команд запросов и ответов каждого из протоколов. Ввиду сложности вариантов 6 и 7 допускается обработка только основного подмножества команд.

Сдача всех программ должна сопровождаться демонстрацией работы не только написанного самостоятельно клиента с сервером, но и демонстрации возможности взаимодействия написанного клиента (сервера) с сервером (клиентом) стороннего производителя (это необходимо для того, что бы проверить что протокол, реализованный самостоятельно, совместим со стандартом).

Лабораторная работа №4

Цель работы: ознакомиться с семейством протоколов прикладного уровня:

1. and D. Reed, May 1993

2. File Transfer Protocol (FTP), RFC 959, by J. Postel and J. Reynolds, October 1985.

3. Hypertext Transfer Protocol version 1.0 (HTTP/1.0), RFC 1945.

4. Internet Relay Chat Protocol (IRC), RFC 1459, by J. Oikarinen Hypertext Transfer Protocol version 1.1 (HTTP/1.1), RFC 2616.

5. The Secure HyperText Transfer Protocol, RFC 2660.

6. Post Office Protocol Version 3, RFC 1939, by J. Myers, May 1996.
7. Simple Mail Transfer Protocol (SMTP), RFC 821.
8. SMTP Service Extensions, RFC 1869.
9. SMTP Service Extension for Authentication, RFC 2554.
10. Network News Transfer Protocol, RFC 977, by Brian Kantor and Phil Fapsley, February 1986.
11. Common NNTP Extensions, RFC 2980, by S. Barber, October 2000.

Теоретические сведения

Протокол HTTP

Hypertext Transfer Protocol (HTTP, протокол пересылки гипертекста) - это язык, которым клиенты и серверы World Wide Web пользуются для общения между собой. Он, по сути дела, является основой в Web. Хотя HTTP в большей степени относится к сфере программирования серверов и клиентов, знание этого протокола важно и для CGI-программирования. Кроме того, иногда HTTP фильтрует информацию и передает ее обратно пользователям - это происходит, например, когда в окне браузера отображаются коды ошибок сервера.

Все HTTP-транзакции имеют один общий формат. Каждый запрос клиента и ответ сервера состоит из трех частей: строки запроса (ответа), раздела заголовка и тела. Клиент инициирует транзакцию следующим образом:

1. Клиент устанавливает связь с сервером по назначенному номеру порта (по умолчанию - 80). Затем клиент посылает запрос документа, указав HTTP- команду, называемую методом, адрес документа и номер версии HTTP. Например, в запросе

GET /index.html HTTP/1.0

используется метод GET, которым с помощью версии 1.0 HTTP запрашивается документ index.html. Методы HTTP более подробно рассматриваются ниже.

2. Клиент посылает информацию заголовка (необязательную), чтобы сообщить серверу информацию о своей конфигурации и данные о форматах документов, которые он может принимать. Вся информация заголовка указывается построчно, при этом в каждой строке приводится имя и значение. Например, приведенный ниже заголовок, посланный клиентом, содержит его имя и номер версии, а также информацию о некоторых предпочтительных для клиента типах документов:

User-Agent: Mozilla/4.05 (WinNT; 1)
*Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */**

Завершается заголовок пустой строкой.

3. Послав запрос и заголовки, клиент может отправить и дополнительные данные. Эти данные используются главным образом теми CGI-программами, которые применяют метод POST. Клиенты также могут использовать их для помещения отредактированной страницы обратно на Web-сервер.

Сервер отвечает на запрос клиента следующим образом:

1. Первая часть ответа сервера - строка состояния, содержащая три поля: версию HTTP, код состояния и описание. Поле версии содержит номер версии HTTP, которой данный сервер пользуется для передачи ответа.

Код состояния - это трехразрядное число, обозначающее результат обработки сервером запроса клиента. Описание, следующее за кодом состояния, представляет собой просто понятный для человека текст, поясняющий код состояния. Например, строка состояния

HTTP/1.0 200 OK

говорит о том, что сервер для ответа использует версию HTTP 1.0. Код состояния 200 означает, что запрос клиента был успешным и затребованные данные будут переданы после заголовков.

2. После строки состояния сервер передает клиенту информацию заголовка, содержащую данные о самом сервере и затребованном документе. Ниже приведен пример заголовка:

Date: Fri, 02 Sep 2022 07:34:28 GMT
Server: Apache/2.2.6
Last-modified: Mon, 29 Aut 2022 22:54:48 GMT
Content-type: text/html
Content-length: 2482

Завершают заголовок 2 пустые строки.

3. Если запрос клиента успешен, то посылаются затребованные данные. Это может быть копия файла или результат выполнения CGI-программы. Если запрос клиента удовлетворить нельзя, передаются дополнительные данные в виде понятного для пользователя разъяснения причин, по которым сервер не смог выполнить данный запрос.

В HTTP L0 за передачей сервером затребованных данных следует разъединение с клиентом, и транзакция считается завершенной, если не передан заголовок Connection: Keep Alive.

Запросы клиента разбиваются на три раздела. Первая строка сообщения всегда содержит HTTP-команду, называемую методом, URI, который обозначает запрашиваемый клиентом файл или ресурс, и номер версии HTTP.

Следующие строки запроса клиента содержат информацию заголовка. Информация заголовка содержит сведения о клиенте и информационном объекте, который он посылает серверу. Третья часть клиентского запроса представляет собой тело содержимого - собственно данные, посылаемые серверу.

Метод - это HTTP-команда, с которой начинается первая строка запроса клиента. Метод сообщает серверу о цели запроса. Для HTTP определены три основных метода: GET, HEAD и POST. Определены и другие методы, но они не так широко поддерживаются серверами, как три перечисленных. При задании имен методов учитывается регистр, поэтому «GET» и «get» различаются.

Метод GET

GET - это запрос информации, расположенной на сервере по указанному URL. GET - наиболее распространенный метод поиска с помощью браузеров документов для визуализации. Результат запроса GET может представлять собой, например, файл, доступный для сервера, результат выполнения программы или CGI-сценария, выходную информацию аппаратного устройства и т.д.

Если клиент пользуется в своем запросе методом GET, сервер отвечает строкой состояния, заголовками и затребованными данными. Если сервер не может обработать запрос вследствие ошибки или отсутствия полномочий, он, как правило, посылает в информационном разделе ответа текстовое пояснение.

Тело информационного содержимого запроса GET всегда пустое. GET в переводе на человеческий язык означает примерно следующее: "Дайте мне этот файл". Для идентификации указанных в запросе клиента файла или программы обычно используется полное имя этого объекта на сервере.

Ниже приведен пример успешного запроса GET на получение файла. Клиент посылает запрос:

GET /index.html HTTP/1.0
Connection: Keep-Alive
User-Agent: Mozilla/4.05 (WinNT; 1)
Host: www.ora.com
*Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */**

Сервер отвечает:

HTTP/1.0 200 Document follows
Date: Fri, 20 Aug 2022 08:17:58 GMT Server: Apache/1.2.6
Last-modified: Mon, 20 Jun 2003 21:53:08 GMT
Content-type: text/html Content-length: 2482

(далее следует тело документа)

Метод POST

Метод POST позволяет посылать на сервер данные в запросе клиента. Эти данные направляются в программу обработки данных, к которой сервер имеет доступ (например, в CGI-сценарий). Метод POST может использоваться во многих приложениях. Например, его можно применять для передачи входных данных для: сетевых служб (таких как телеконференции); программ с интерфейсом в виде командной строки; аннотирования документов на сервере; выполнения операций в базах данных.

Данные, посылаемые на сервер, находятся в теле содержимого запроса клиента. По завершении обработки запроса POST и заголовков сервер передает тело содержимого в программу, заданную URL. В качестве схемы кодирования с методом POST используется Ba8e64-кодирование, которое позволяет преобразовывать данные форм в список переменных и значений для CGI-обработки. Ниже приведен небольшой пример запроса клиента с использованием метода POST. Клиент посылает на сервер данные о дне рождения, введенные в форму:

POST /cgi-bin/birthday.pl HTTP/1.0
User-Agent; Mozilla/4.05 (WinNT; 1)
*Accept: image/gif, image/x-xbj.tmap, image/jpeg, J.mage/pjpeg, */**
Host: www.ora.com
Content-type: application/x-www-form-ur.lencoded
Content-Length: 20
nionth=august&date=24
Ответы сервера

Ответ сервера на запрос клиента состоит из трех частей. Первая строка - это строка ответа сервера, которая содержит номер версии HTTP, число, обозначающее состояние запроса, и краткое описание состояния. После строки ответа следует информация заголовка и тело содержимого, если таковое имеется.

Протокол SMTP

Основная задача протокола ***SMTP*** (***Simple Mail Transfer Protocol***) заключается в том, чтобы обеспечивать передачу электронных сообщений (почту). Для работы через протокол SMTP клиент создаёт TCP соединение с сервером через порт 25. Затем клиент и SMTP сервер обмениваются информацией пока соединение не будет закрыто или прервано. Основной процедурой в SMTP является передача почты (Mail Procedure). Далее идут процедуры форвардинга почты (Mail Forwarding), проверка имён почтового ящика и вывод списков почтовых групп. Самой первой процедурой является

открытие канала передачи, а последней - его закрытие.

При этом отправитель инициирует соединение и посылает запросы на обслуживание, а получатель - отвечает на эти запросы. Фактически отправитель выступает в роли клиента, а получатель - сервера. Канал связи устанавливается непосредственно между отправителем и получателем сообщения. При таком взаимодействии почта достигает абонента в течение нескольких секунд после отправки

Команды SMTP указывают серверу, какую операцию хочет произвести клиент. Команды состоят из ключевых слов, за которыми следует один или более параметров. Ключевое слово состоит из 4-х символов и разделено от аргумента одним или несколькими пробелами. Каждая командная строка заканчивается символами перевода строки (CRLF). Вот синтаксис всех команд протокола SMTP (SP - пробел):

HELO <SP> <domain> <CRLF>
MAIL <SP> FROM:<reverse-path> <CRLF>
RCPT <SP> TO:<forward-path> <CRLF>
DATA <CRLF>
RSET <CRLF>
SEND <SP> FROM:<reverse-path> <CRLF>
SOML <SP> FROM:<reverse-path> <CRLF>
SAML <SP> FROM:<reverse-path> <CRLF>
VRFY <SP> <string> <CRLF>
EXPN <SP> <string> <CRLF>
HELP <SP> <string> <CRLF>
NOOP <CRLF>
QUIT <CRLF>

Обычный ответ SMTP сервера состоит из номера ответа, за которым через пробел следует дополнительный текст. Номер ответа служит индикатором состояния сервера.

Отправка почты

Первым делом подключаемся к SMTP серверу через порт 25. Теперь надо передать серверу команду HELLO и наш IP адрес (здесь и далее символически обозначены: C: - запрос клиента, S: - ответ сервера):

C: HELLO 195.161.101.33 S: 250

smtp.mail.ru is ready

При отправке почты передаём некоторые нужные данные (отправитель, получатель и само письмо):

C: MAIL PROM: <отправитель>
S: 250 OK

C: RCPT TO: <получатель>

S: 250 OK

указываем серверу, что будем передавать содержание письма (заголовок и тело письма)

C: DATA

S: 354 Start mail input; end with <CRLL>.<CRLP>

<само письмо>

передачу письма необходимо завершить символами CRLL.CRLL

S: 250 OK

Теперь завершаем работу, отправляем команду QUIT:

S: QUIT

C: 221 smtp.mail.m is closing transmission channel

Пример:

C: Prom: Drozd <vasya@tut.by>

C: To: Drol <petya@mail.ru >

C: Subject: Hello C: Hello vasya!

C: How are you?

<CRLP.CRLP>

S: 250 OK S: QUIT

C: 221 smtp.mail.m is closing transmission channel

Протокол POP3

POP3 - это простейший протокол для работы пользователя с содержимым своего почтового ящика. Он позволяет только забрать почту из почтового ящика сервера на рабочую станцию клиента и удалить ее из почтового ящика на сервере. Всю дальнейшую обработку почтовое сообщение проходит на компьютере клиента.

POP3-сервер не отвечает за отправку почты, он работает только как универсальный почтовый ящик для группы пользователей. Когда пользователю необходимо отправить сообщение, он должен установить соединение с каким-либо SMTP-сервером и отправить туда свое сообщение по SMTP. Этот SMTP сервер может быть тем же хостом, где работает POP3-сервер, а может располагаться совсем в другом месте.

POP3-сервис, как правило, устанавливается на 110-й TCP-порт сервера, который будет находиться в режиме ожидания входящего соединения. Когда клиент хочет воспользоваться POP3 -сервисом, он просто устанавливает TCP-соединение с портом 110 этого хоста. После установления соединения сервис

POP3 отправляет подсоединившемуся клиенту приветственное сообщение. После этого клиент и сервер начинают обмен командами и данными. По окончании обмена POP3-канал закрывается.

Команды POP3 состоят из ключевых слов, состоящих из ASCII-символов, и одним или несколькими параметрами, отделяемыми друг от друга символом "пробела" - <SP>. Все команды заканчиваются символами "возврата каретки" и "перевода строки" - <CRLF>. Длина ключевых слов не превышает четырех символов, а каждого из аргументов может быть до 40 символов.

Ответы POP3-сервера на команды состоят из строки статус-индикатора, ключевого слова, строки дополнительной информации и символов завершения строки - <CRLF>. Длина строки ответа может достигать 512 символов. Строка статус-индикатора принимает два значения: положительное ("+OK") и отрицательное ("-ERR"). Любой сервер POP3 обязан отправлять строки статус-индикатора в верхнем регистре, тогда как другие команды и данные могут приниматься или отправляться как в нижнем, так и в верхнем регистрах.

Ответы POP3-сервера на отдельные команды могут составлять несколько строк. В этом случае строки разделены символами <CRLF>. Последнюю строку информационной группы завершает строка, состоящая из символа "." (код - 046) и <CRLF>, т.е. последовательность "CRLF.CRLF".

POP3-сессия состоит из нескольких частей. Как только открывается TCP-соединение и POP3-сервер отправляет приветствие, сессия должна быть зарегистрирована - состояние аутентификации (AUTHORIZATION state). Клиент должен зарегистрироваться в POP3-сервере, т. е. ввести свой идентификатор и пароль.

После этого сервер предоставляет клиенту его почтовый ящик и открывает для данного клиента транзакцию - состояние начала транзакции обмена (TRANSACTION state). На этой стадии клиент может считать и удалить почту своего почтового ящика.

После того как клиент заканчивает работу (передает команду QUIT), сессия переходит в состояние UPDATE - завершение транзакции. В этом состоянии POP3-сервер закрывает транзакцию данного клиента (на языке баз данных - операция COMMIT) и закрывает TCP-соединение.

В случае получения неизвестной, неиспользуемой или неправильной команды, POP3-сервер должен ответить отрицательным состоянием индикатора.

При открытии TCP-соединения POP3-клиентом, POP3-сервер отправляет сообщение приветствия (далее во всех примерах POP3-протокола используются следующие обозначения: C - клиент, S - сервер POP3):

S: +OK POP3 server ready

Теперь POP3-сессия находится в состоянии авторизации (AUTHORIZATION), и клиент должен зарегистрировать себя на POP3-сервере. Это может быть выполнено либо с помощью команд USER и PASS - ввод открытых пользовательского идентификатора и пароля (именно этот способ используется чаще), либо командой APOP - авторизации цифровой подписью, на базе секретного ключа. Любой POP3-сервер должен поддерживать хотя бы один из механизмов авторизации.

Команда USER имеет следующий формат:

USER name

Аргументом (name) является строка, идентифицирующая почтовый ящик системы. Этот идентификатор должен быть уникальным в данной почтовой системе POP3-сервера. Если ответом на эту команду является строка индикатора "+OK", клиент может отправлять команду PASS - ввод пароля или QUIT - завершить сессию. Если ответом является строка "-ERR", клиент может либо повторить команду USER, либо закрыть сессию. Примеры использования команды:

C: USER frated

S: -ERR sorry, no mailbox for frated here

или

C: USER mrose

S: +OK mrose is a real hoopy frood

Команда PASS используется только после положительного ответа на команду USER:

PASS string

Аргументом команды является строка пароля данного почтового ящика. После получения команды PASS, POP3-сервер, на основании аргументов команд USER и PASS, определяет возможность доступа к заданному почтовому ящику. Если POP3-сервер ответил "+OK", это означает, что авторизация клиента прошла успешно и он может работать со своим почтовым ящиком, т.е. сессия переходит в состояние TRANSACTION. Если POP3-сервер ответил "-ERR", то либо был введен неверный пароль, либо не найден указанный почтовый ящик:

C: USERmrose

S: +OK mrose is a real hoopy frood

C: PASS secret

S: +OK mrose's maildrop has 2 messages (320 octets)

Команда закрытия POP3-сессии: QUIT - отправляется без аргументов и всегда имеет единственный ответ "+OK", например:

C: QUIT

S: +OK dewey POP3 server signing off

После того как клиент успешно прошел процедуру авторизации в POP3-сервере, и POP3-сервер "закрыл" определенный почтовый ящик только для использования данным клиентом (для тех, кто работал с базами данных, это называется EXCLUSIVE ACCESS LOCK), POP3-сессия переходит в режим TRANSACTION, и клиент может начать работу со своей почтой.

Команда ST AT (без аргументов) используется для просмотра состояния текущего почтового ящика.

В ответ POP3- сервер возвращает строку, содержащую количество и общий размер в байтах сообщений, которые клиент может получить с POP3-сервера. Сообщения, помеченные на удаление, не учитываются. Формат ответа: "+OK nn mm", где nn - количество сообщений, mm - их общий объем:

C: STAT

S: +OK2 320

Команда RETR - используется для передачи клиенту запрашиваемого сообщения:

RETR msg

Аргумент команды - номер сообщения. Если запрашиваемого сообщения нет, возвращается отрицательный индикатор "-ERR".

C: RETR 1

S: +OK 120 octets

S: <text message>

S: .

После получения, сообщение, как правило, помечается на удаление из почтового ящика, при этом используется команда DELE:

DELE msg

Аргумент команды: номер сообщения. Сообщения, помеченные на удаление, реально удаляются только после закрытия транзакции, на стадии UPDATE.

C: DELE 1

S: +OK message 1 deleted ИЛИ

C: DELE 2

S: -ERR message 2 already deleted

Для проверки состояния соединения с POP3-сервером используется команда NOOP. При активном соединении ответом на нее будет положительный индикатор "+OK":

C: NOOP

S: +OK

Ниже приведена стандартная сессия работы с POP3 -протоколом.

S: <wait for connection on TCP port 110>
C: <open connection>
S: +OK POP3 server ready
C: USER mrose
S: +OK mrose is a real hoopy frood
C: PASS secret
S: +OK mrose's maildrop has 2 messages (320 octets)
C: STAT
S: +OK 2 320
C: LIST
S: +OK 2 messages (320 octets)
S: 1 120
S: 2 200
S: .
C: RETR 1
S: +OK 120 octets
S: <the POP3 server sends message 1>
S: .
C: DELE 1
S: +OK message 1 deleted
C: RETR 2
S: +OK 200 octets
S: <the POP3 server sends message 2>
S: .
C: DELE 2
S: +OK message 2 deleted
C: QUIT
S: +OK dewey POP3 server signing off (maildrop empty)
C: <close connection>
S: <wait for next connection>

Простота протокола POP, которая послужила росту его популярности вначале, обернулась затем отсутствием гибкости и невозможности выполнять другие операции управления почтовыми ящиками. На смену POP3 пришло новое поколение протоколов работы с электронной почтой - протоколы IMAP.

Протокол FTP

FTP (RFC-959) обеспечивает файловый обмен между удаленными пользователями. Протокол FTP формировался многие годы. Первые реализации в МТИ относятся к 1971. (RFC 114 и 141). RFC 172

рассматривает протокол, ориентированный на пользователя, и предназначенный для передачи файлов между ЭВМ. Позднее в документах RFC 265 и RFC 281 протокол был усовершенствован. Заметной переделке протокол подвергся в 1973, и окончательный вид он обрел в 1985 году. Таким образом, данный протокол является одним из старейших.

Работа FTP на пользовательском уровне содержит несколько этапов:

1. Идентификация (ввод имени-идентификатора и пароля).
2. Выбор каталога.
3. Определение режима обмена (поблочный, поточный, ascii или двоичный).
4. Выполнение команд обмена (get, mget, dir, mdel, input или put).
5. Завершение процедуры (quit или close).

Команды FTP приведены в табл 4.1.

Таблица 4.1. – команды FTP

Команда	Описание
ABOR	прервать предыдущую команду FTP и любую передачу данных
LIST список файлов	список файлов или директорий
PASS пароль	пароль на сервере
PORT n1,n2,n3,n4,n5,n6	IP адрес клиента (n1.n2.n3.n4) и порт (n5 x 256 + n6)
QUIT	закрыть бюджет на сервере
RETR имя файла	получить (get) файл
STOR имя файла	положить (put) файл
SYST 1	сервер возвращает тип системы
TYPE тип	указать тип файла: A для ASCII, I для двоичного
USER имя пользователя	имя пользователя на сервере

Команды и отклики передаются по управляющему соединению между клиентом и сервером в формате NVT ASCII. В конце каждой строки команды или отклика присутствует пара CR, LF. Команды состоят из 3 или 4 байт, а именно из заглавных ASCII символов, некоторые с необязательными аргументами. Клиент может отправить серверу более чем 30 различных FTP команд.

FTP отклики

Отклики состоят из 3-циферных значений в формате ASCII, и необязательных сообщений, которые следуют за числами. Подобное

представление откликов объясняется тем, что программному обеспечению необходимо посмотреть только цифровые значения, чтобы понять, что ответил процесс, а дополнительную строку может прочитать человек. Поэтому пользователю достаточно просто прочитать сообщение (причем нет необходимости запоминать все цифровые коды откликов). Группы откликов представлены в табл. 4.2.

Таблица 4.2. – отклики FTP

Отклик	Описание
lyz	Положительный предварительный отклик. Действие началось, однако необходимо дождаться еще одного отклика перед отправкой следующей команды.
2yz	Положительный отклик о завершении. Может быть отправлена новая команда.
3yz	Положительный промежуточный отклик. Команда принята, однако необходимо отправить еще одну команду.
4yz	Временный отрицательный отклик о завершении. Требуемое действие не произошло, однако ошибка временная, поэтому команду необходимо повторить позже.
5yz	Постоянный отрицательный отклик о завершении. Команда не была воспринята и повторять ее не стоит.
x0z	Синтаксическая ошибка.
xlz	Информация.
x2z	Соединения. Отклики имеют отношение либо к управляющему, ли-
x3z	Аутентификация и бюджет. Отклик имеет отношение к
x4z	Не определено.
x5z	Состояние файловой системы.

Третья цифра дает дополнительное объяснение сообщению об ошибке. Ниже приведены некоторые типичные отклики с возможными объясняющими строками.

- 125 Соединение данных уже открыто; начало передачи.
- 200 Команда исполнена.
- 214 Сообщение о помощи (для пользователя).
- 331 Имя пользователя принято, требуется пароль.
- 425 Невозможно открыть соединение данных.
- 452 Ошибка записи файла.
- 500 Синтаксическая ошибка (неизвестная команда).

- 501 Синтаксическая ошибка (неверные аргументы).
- 502 Нереализованный тип MODE.

Управление соединением

Использовать соединение данных можно тремя способами.

1. Отправка файлов от клиента к серверу.
2. Отправка файлов от сервера к клиенту.
3. Отправка списка файлов или директорий от сервера к клиенту.

Задания

Все программы ко всем заданиям выполняются на любом языке программирования, допускается использование компонент ClientSocket, ServerSocket, TcpClient, TcpServer в C++Builder/Delphi и аналогичных классов в C++/C#.

РАЗДЕЛ 3. КОНРОЛЬ ЗНАНИЙ

Общая формулировка заданий к контрольной работе

Методические указания по выполнению контрольной работы по дисциплине «Компьютерные системы и сети» студент должен выполнить контрольную работу, которая предоставляется на кафедру и защищается студентом заочной (дистанционной) формы получения образования до начала лабораторно зачётной сессии. При выполнении контрольных работ необходимо соблюдать следующие правила:

1. Контрольная работа может выполняться как в рукописном, так и печатном виде, на титульном листе необходимо указать наименование дисциплины, фамилию и инициалы студента, выполненный вариант (соответствует последним двум цифрам зачетной книжки), шифр специальности и номер группы.

2. Контрольную работу следует выполнять аккуратно, оставляя поля для замечаний рецензента.

3. Для пояснения выполнения работы, где это необходимо, сделать скриншот.

4. Выполнение работы сопровождается пояснениями в виде текстовой информации от разработчика.

5. В пояснениях к задаче необходимо указывать используемые подходы, а так же пояснение к предложенному методу написания программы.

6. Особое внимание уделяется написанию вывода по работе, который должен подчеркивать проделанную работу разработчиком и приобретенные им знания.

7. В контрольной работе следует указывать учебники и учебные пособия, которые использовались при решении поставленных задач.

Задания к контрольной работе

Вариант 1

Написать программу, реализующую функции HTTP-сервера версии 1.0. В обязательном порядке должны поддерживаться следующие виды запросов: GET, POST, HEAD, а так же наиболее распространенные коды ответов. Сервер конфигурируется на определенный каталог, где расположены html-файлы и другие подкаталоги. В главном окне сервера расположено поле типа memo, в котором отображается весь протокол общения HTTP клиента с HTTP сервером, например:

GET /index.html HTTP/1.0
Connection: Keep-Alive
User-Agent: Mozilla/4.05 (WinNT; 1)
Host: www.ora.com
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
HTTP/1.0 200 Document follows
Date: Fri, 28 Jan 2022 08:17:58 GMT Server: Apache/1.2.6
Last-modified: Mon, 26 Jun 2021 21:53:08 GMT
Content-type: text/html
Content-length: 2482
<html><body>

Тестирование и подача программы-сервера производится при помощи Web-браузера.

Вариант 2

Написать программу, реализующую функции HTTP-клиента версии 1.0. В обязательном порядке должны поддерживаться следующие виды запросов: GET, POST, HEAD, а так же наиболее распространенные коды ответов. Отображение полученных данных в форматированном виде не обязательно (можно в виде plain text). В окне клиента должно быть расположено поле типа memo в котором отображается весь протокол общения HTTP клиента с HTTP сервером (см. задание 1). Тестирование и подача HTTP клиента производится при помощи запроса к реальному web-серверу, расположенному в Internet или установленному в локальной сети, или при помощи запроса к web-серверу, написанному в предыдущем задании.

Вариант 3

Написать программу, реализующую функции POP3-сервера. В главном окне сервера расположено поле типа memo, в котором отображается весь про¬токол общения клиента с сервером, например:

STAT
+OK 2 320

LIST

+OK 2 messages (320 octets)

Тестирование и подача программы-сервера производится при помощи любого стандартного почтового клиента.

Вариант 4

Написать программу, реализующую функции POP3-клиента. В главном окне клиента расположено поле типа memo, в котором отображается весь протокол общения клиента с сервером (см. задание 3).

Тестирование и подача программы-клиента производится при помощи любого стандартного почтового сервера, расположенного в сети Internet или локальной сети. В качестве сервера может использоваться программа, написанная к заданию 3.

Вариант 5

Написать программу, реализующую функции SMTP-сервера. В главном окне сервера расположено поле типа memo, в котором отображается весь протокол общения клиента с сервером, например:

MAIL From <Alex@mail.ru>

250 Sender ok

RCPT To:<ysemenov@mail.ch>

250 <ysemenov@mail.ch> Recepient ok

DATA

Тестирование и подача программы-сервера производится при помощи любого стандартного почтового клиента.

Вариант 6

Написать программу, реализующую функции SMTP-клиента. В главном окне клиента расположено поле типа memo, в котором отображается весь проокол общения клиента с сервером (см. задание 5).

Тестирование и подача программы-клиента производится при помощи любого стандартного почтового сервера, расположенного в сети Internet или локальной сети. В качестве сервера может использоваться программа, написанная к заданию 5.

Вариант 7

Написать программу, реализующую функции FTP-сервера. В главном окне сервера расположено поле типа memo, в котором отображается весь протокол общения клиента с сервером, например:

USER vasilisa

331 Password required for vasilisa.

PASS abed

230 User vasilisa logged in.

PORT 140,252,13,34,4,150

200 PORT command successful.

…

Тестирование и подача программы-сервера производится при помощи любого стандартного FTP клиента.

Вариант 8

Написать программу, реализующую функции FTP-клиента. В главном окне клиента расположено поле типа memo, в котором отображается весь протокол общения клиента с сервером (см. задание 7).

Тестирование и подача программы-клиента производится при помощи любого стандартного FTP сервера, расположенного в сети Internet или локальной сети. В качестве сервера может использоваться программа, написанная к заданию 7.

РАЗДЕЛ 4. ВСПОМОГАТЕЛЬНЫЙ

ПРОГРАММА ДИСЦИПЛИНЫ

Учебная программа по учебной дисциплине «Компьютерные системы и сети» разработана для специальности 1-40 01 01 «Программное обеспечение информационных технологий» специализации 1-40 01 01 01 «Веб-технологии и программное обеспечение мобильных систем».

Подготовка современного специалиста требует уверенного владения возможностями, предоставляемыми компьютерными технологиями. Изучение настоящей дисциплины обеспечивает подготовку специалиста, владеющего фундаментальными знаниями и практическими навыками в области компьютерных сетей и сетевого программирования.

Цели дисциплины:

- теоретическая и практическая подготовка, обеспечивающая получение знаний по основам компьютерных сетей;
- получения практических навыков программирования сетевых протоколов;
- получение навыков проектирования компьютерных сетей.

Задачи дисциплины:

- подготовка специалиста, имеющего устойчивые навыки использования локальных и глобальных компьютерных сетей;
- формирование базовых навыков проектирования компьютерных сетей, эффективного использования и настройки сетевого оборудования;
- формирование навыков программирования сетевых технологий.

Для изучения дисциплины «Компьютерные системы и сети» необходимы знания, получаемые при изучении дисциплины «Основы алгоритмизации и программирования». В свою очередь учебная дисциплина «Компьютерные системы и сети» является базой для таких учебных дисциплин, как «Программирование сетевых приложений», «Визуальные средства разработки программных приложений», «Распределенные информационные системы».

Освоение данной учебной дисциплины обеспечивает формирование следующих компетенций:

Для специальности 1-40 01 01 «Программное обеспечение информационных технологий» (срок обучения – 4 года):

БПК-15. Использовать общепринятые подходи и построения, конфигурирования и администрирования компьютерных систем и сетей.

Для специальности 1-40 01 01 «Программное обеспечение информационных технологий» (срок обучения – 5 лет):

БПК-15. Использовать общепринятые подходи и построения, конфигурирования и администрирования компьютерных систем и сетей.

В результате изучения учебной дисциплины студент должен:

знать:

- основные концепции построения локальных и глобальных сетей; методы объединения компьютеров и устройств в сети;
- основные функции и режимы взаимодействия компьютеров, аппаратное и программное обеспечение сети;
- основные протоколы, методы организации, способы объединения компьютеров в сети;
- виды топологий сети и основные реализуемые алгоритмы взаимодействия узлов;
- способы передачи, методы кодирования и защиты данных;
- принципы разработки программ организации клиент-серверного взаимодействия, методы разработки программ распределенной обработки данных;
- перспективные направления развития компьютерных сетей и сетевых технологий, методы использования сетей и сетевых технологий в будущей профессиональной деятельности;

уметь:

- анализировать уровень эффективности сетевых решений;
- эффективно использовать операционные системы и предлагать сетевые решения для разрабатываемых прикладных задач;
- разрабатывать программы взаимодействия для работы в архитектуре клиент сервер для организации клиент-серверного взаимодействия и распределенной обработки данных;
- использовать различные протоколы при разработке программных средств;

владеть:

- методами разработки и обоснования конфигурации сети, оценки трафика в сегментах, выбором сетевого оборудования и программного обеспечения;
- техникой конфигурирования локальных сетей, реализации сетевых протоколов с помощью программных средств;
- базовыми методами и программными средствами разработки сетевых приложений;
- методиками постановки и решения задачи проектирования или

модернизации локальной или корпоративной вычислительной сети;
- навыками работы с информацией в локальных и глобальных компьютерных сетях.

Согласно учебному плану для заочной (дистанционной) формы получения высшего образования (срок обучения – 4 и 5 лет) на изучение учебной дисциплины отведено всего 196 ч., из них аудиторных – 30 часов.

Распределение аудиторных часов по курсам, семестрам и видам занятий приведено в таблице 1.

Таблица 1.

Заочная (дистанционная) форма получения высшего образования (срок обучения – 4 и 5 лет)					
Курс	Семестр	Лекции, ч.	Лабораторные занятия, ч.	Практические занятия, ч.	Форма текущей аттестации
2	4	16	14		экзамен

СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Раздел 1. ОБЩИЕ ПРИНЦИПЫ ПОСТРОЕНИЯ КОМПЬЮТЕРНЫХ СЕТЕЙ

Тема 1.1. Определение компьютерной сети. Обобщенная схема функционирования сети

Телекоммуникация, коммуникационная сеть, информационная сеть, вычислительная сеть. Компьютерная сеть (определение, назначение, цель использование). Предпосылки и причины появления сетей. Обобщенная схема функционирования сети.

Тема 1.2. Классификация, характеристики компьютерных сетей

Локальные, корпоративные, региональные и глобальные компьютерные сети. Особенности построения и функционирования, отличия. Конвергенция сетей.

Тема 1.3. Понятие протокола и применение сетевых протоколов для взаимодействия объектов сети

Основные принципы построения сети. Многоуровневый подход к решению задачи обмена сообщениями между компьютерами. Основные понятия о протоколе. Стек протоколов. Модель OSI.

Тема 1.4. Требования, предъявляемые к современным сетям

Требования, предъявляемые к современным вычислительным сетям.

Проблемные ситуации, возникающие в различных типах сетей, методы и средства их решения. Производительность, надежность и безопасность. Расширяемость и масштабируемость. Прозрачность, управляемость и совместимость.

Раздел 2. ЛОКАЛЬНЫЕ КОМПЬЮТЕРНЫЕ СЕТИ

Тема 2.1. Классификация локальных сетей

Сети с централизированным управлением, иерархические сети: одноранговые и с выделенным сервером (сравнительный анализ, области применения). Технология клиент-сервер. Виды серверов.

Тема 2.2. Топологии локальных сетей: Физическая и логическая.

Понятие топологии при построении компьютерных сетей. Логическая и физическая топологии сети. Топология шина, особенности реализации, коллизия, разделение передающей среды, надежность, безопасность, стоимость реализации. Передающая среда для построения сети по топологии звезда, ограничения, стоимость и безопасность реализации сети. Топологии, в которых отсутствуют коллизии. Особенности реализации топологии кольцо, стоимость и безопасность. Сотовая, полносвязная, древовидная и петлевая топологии, как производные топологии, основанные на трех базовых. Области их использования, примеры.

Тема 2.3. Среда передачи: проводная и беспроводная. Коаксиальный кабель, витая пара, оптоволокно. Радиоволны, микроволны, инфракрасное излучение

Проводная и беспроводная среда передачи. Коаксиальный кабель, как основная среда для реализации сети по топологии шина. Основные конструктивные элементы, помехозащищенность, технологичность, проблемы обслуживания и монтажа, стоимость. Витая пара, как основная среда для построения сети по топологии звезда. Категории витой пары, отличия, конструктивные элементы, помехозащищенность, ограничения и стоимость реализации. Принцип функционирования оптических сред передачи даны. Одномодовый и многомодовый (с линейным и градиентным коэффициентом преломления) кабель. Скорости, особенности монтажа, расстояния, модернизация, стоимость и безопасность реализации сети на базе оптоволоконного кабеля. Радиосети. Радиорелейные сети. Спутниковая связь. Сети транкинговой связи. Инфракрасные беспроводные сети, скорости, расстояния и особенности реализации. Структура, классификация, протоколы систем мобильной связи.

Методы передачи данных на физическом уровне. Основы кодирование сигналов. Физическое кодирование. Потенциальное и импульсное кодирование. Аналоговая модуляция и методы аналоговой модуляции. Цифровое кодирование. Логическое кодирование. Дискретная модуляция аналоговых сигналов.

Тема 2.4. Методы доступа к среде передачи: конфликтные и бесконфликтные

Классификация методов доступа к среде передачи. Метод доступа CSMA/CD. Метод доступа CSMA/CA. Метод доступа приоритету. Маркерные методы доступа.

Тема 2.5. Модель взаимодействия открытых систем. Стеки протоколов

Многоуровневая модель OSI, модель и взаимодействие протоколов. Примеры протоколов. Сетевые протоколы. Стеки протоколов.

Тема 2.6. Базовые технологии локальных сетей

Стандарты локальных сетей. Подуровни канального уровня модели OSI. История появления и характеристика сетей Ethernet. Ограничения и правила построения сетей Ethemet. Расчет времени оборачиваемости сигнала и сокращение межкадрового расстояния. Преодоление ограничений топологий на основе произведенных расчетов. Особенности выбора оборудования и комбинации производных топологий для оптимального функционирования сети. Коммутируемые сети Ethemet. Скоростные версии Ethemet. Сетевые технологии локальных сетей: 100VG AnyLan, ArcNet, Token Ring, FDDI. Ограничения и правила построения кольцевых сетей.

Раздел 3. ОБЪЕДИНЕНИЯ СЕТЕЙ И ГЛОБАЛЬНЫЕ СЕТИ

Тема 3.1. Принципы межсетевого взаимодействия

Гетерогенность и проблемы межсетевого взаимодействия. Основные подходы к организация межсетевого взаимодействия. Мультиплексирование стеков протоколов. Место размещения средств межсетевого взаимодействия. Особенности согласования сетей на транспортном уровне. Источники и типы неоднородностей в транспортной подсистеме. Средства согласования физического уровня. Средства согласования на канальном уровне. Сетевые устройства: повторители, концентраторы, мосты, коммутаторы, маршрутизаторы.

Тема 3.2. Сети tcp/ip

Принципы объединения сетей с помощью протоколов сетевого уровня. Семейство протоколов TCP/IP. Транспортные протоколы TCP и UDP. Протокол межсетевого взаимодействия IP, версии протокола. Адресация в IP-сетях. Использование масок и подсетей. Разрешение IP адресов в Ethernet сетях. Маршрутизация IP-адресов. Фрагментация IP-пакетов. Типы протоколов обмена маршрутной информацией. Протоколы DHCP, OSPF, RIP, ARP, RARP. Протокол ICMP. IPv6 как развитие стека TCP/IP.

Тема 3.3. Глобальные сети и перспективные сетевые технологии

Методы коммутации. Коммутация каналов. Коммутация сообщений. Коммутация пакетов. Мультиплексирование, виды мультиплексирования. Плезиохронная и синхронная цифровые иерархии. Передача данных по выделенным линиям. Построение компьютерных сетей на основе телефонных сетей с коммутацией каналов. Сети ISDN. Компьютерные глобальные сети с коммутацией пакетов. Сети X.25, Frame Relay. Технология ATM, основные принципы технологии ATM, стек протоколов ATM, классы сервиса. Обобщенная структура телекоммуникационной сети. Сеть доступа. Транспортная сеть. Сетевое управление. Сетевой интеллект.

Тема 3.4. Глобальная сеть интернет

История возникновения и развития. Определение. Принципы построения глобальной компьютерной сети Интернет. Сервисы сети Интернет. Всемирная паутина. URL. Протокол HTTP. Электронная почта. Протоколы электронной почты, почтовые клиенты, безопасность. Протокол передачи файлов. Сетевое управление в IP-сетях.

ТРЕБОВАНИЯ К КУРСОВОМУ ПРОЕКТУ

Целью курсового проекта является выработка и закрепление практических навыков по выполнению задач проектирования сетей передачи данных с применением современного оборудования.

Курсовой проект преследует цели повышения качеств и углубления знаний студентов в области планирования и распределения сетевых элементов единой сети передачи данных. Задания проекта затрагивают аспекты оптимизации программно- аппаратного ресурса сети, эффективного использование доступного адресного пространства, построение оптимальной структуры резервных связей подсетей и т.д. Во второй части работы, посвященной развертыванию ядра компьютерной сети с помощью специализированного программного комплекса проводится поэтапная апробация теоретических расчетов и разработанного плана распределения сети.

Содержание, состав, объем, и структурное построение курсовых проектов зависят от их типа и специфики темы и должны соответствовать утвержденному заданию. Объем пояснительной записки к курсовому проекту должен составлять 35 - 40 страниц (но не более 40) страниц печатного текста.

Общими требованиями к пояснительной записке являются: четкость логическая последовательность изложения материала, убедительность аргументации, краткость и ясность формулировок, исключающих неоднозначность толкования, конкретность изложения результатов, доказательств и выводов.

Ответственность за достоверность полученных результатов, принятых

решений и выводов в работе несет разработчик.

В соответствии с учебным планом на выполнение курсового проекта отводится всего 60 часов, в том числе 30 аудиторных часов на лекционные и лабораторные занятия для студентов срок обучения – 4 года и 16 аудиторных часов на лекционные и лабораторные занятия для студентов срок обучения – 5 лет.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА

Номер раздела, темы	Название раздела, темы, занятия	Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия
1	2	3	4	5	6
	3 семестр				
1.	Общие принципы построения компьютерных сетей				
1.1	Определение компьютерной сети. Обобщенная схема функционирования сети	2			
1.2	Классификация, характеристики компьютерных сетей	2			
2	Локальные компьютерные сети				
2.1	Классификация, локальных сетей	2			
2.2	Топологии локальных сетей: физическая и логическая.	2			
	Лабораторное занятие №1 Принципы работы сети				2
	Лабораторное занятие №2 Архитектура прикладного интерфейса Windows Sockets				2
	Лабораторное занятие №3 Протоколы Echo, Time, DayTime, Wliols, Finger, RLogin, Telnet				4
3	Объединения сетей и глобальные сети				
3.1	Принципы межсетевого взаимодействия	2			
3.2	Сети TCP/IP	2			
3.3	Глобальные сети и перспективные сетевые технологии	4			
	Лабораторное занятие №4 Семейство протоколов прикладного уровня				6
	Итого	14			14

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

ЛИТЕРАТУРА

Основная литература

1. Олифер, В.Г. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов / В.Г. Олифер, Н.А. Олифер. - 6-е изд. - СПб: Питер, 2020. - 1008с.

2. Таненбаум, Э. Компьютерные сети/ Э. Таненбаум, Д. Уззеролл. - 5-е изд. - СПб: Питер, 2016. - 960с.

3. Куроуз, Дж. Компьютерные сети: Нисходящий подход/ Джеймс Куроуз, Кит Росс.- 6е изд.- М: «Эксмо», 2016.- 912с.

4. Закер. К. Компьютерные сети. Модернизация и поиск неисправностей: Пер. с англ. - СПб.: БХВ-Петербург, 2004. - 1008 с.: ил.

5. MSDN Library. Раздел Networking and Directory Services - Network Protocols - NetBIOS, раздел Networking and Directory Services - Network Management - Windows Networking (WNet).

6. Windows: Сети: Изучение TCP/IP. [Электрон, ресурс] / OS Zone. Режим доступа: http://www.oszone.net/display.php7idM409

7. Request for comment. [Электрон, ресурс] / Режим доступа: http://www.rfc-editor.org/rfc/

Дополнительная литература

8. Одом, У. Официальное руководство Cisco по подготовке к сертификационным экзаменам CCENT/CCNA ICND1 100-105/ Уэнделл Одом.- М.: Изд-во Вильямс, 2018. – 1088с.

9. Глейзер, Дж. Многопользовательские игры. Разработка сетевых приложений/ Дж. Глейзер, С. Мадхав. - СПб: Питер, 2017. – 368с.

10. Леонтьев, В. Windows 10/ В. Леонтьев. – 4-е изд. - М: «Эксмо», 2019.- 384с.

11. Куроуз, Дж. Компьютерные сети. Настольная книга системного администратора/ Джеймс Куроуз, Кит Росс.- 6-е изд.- М: «Эксмо», 2016.- 912с.

12. Робачевский, А. Интернет изнутри. Экосистема глобальной сети/ А. Робачевский.- 2-е изд. – М.: Изд-во Альпина Паблишер, 2017. – 224с.

13. Сергеев, А. Основы локальных компьютерных сетей. Учебное пособие/ А. Сергеев. – СПб: Изд-во Лань, 2016. – 184с.

14. Стандарты Интернета (RFC) [Электронный ресурс].- Режим доступа: http://rfc.com.ru/

15. Справочник по командам Windows. [Электрон, ресурс] / Режим доступа: http://winclianger.wliatis.ru/file/prog.zip

16. Паркер Т., Сиян К. TCP/IP. Для профессионалов.: Пер. с англ. - СПб.: Питер, 2003. 864 с.: ил.

17. Протоколы: HTTP, FTP, RPC, SMTP, POP3, IMAP4, SNMP, IPX/SPX - Книга. [Электрон, ресурс] / Исходники.RU. Режим доступа: http://ishodniki.m/list/?show=protocols

18. Паркер Т., Сиян К. TCP/IP. Для профессионалов.: Пер. с англ. - СПб.: Питер, 2003. 864 с.: ил.

Перечень тем курсовых проектов

1. Сеть на основе «витой пары».
2. Сеть на основе оптоволоконного кабеля.
3. Сеть на основе беспроводных технологий (WiFi).
4. Сеть на основе беспроводных технологий (спутниковой передачи данных).
5. Логическая группировка сети по типу рабочей группы.
6. Логическая группировка сети по типу доменной зоны.
7. Сеть с использованием маршрутизации.
8. Сеть с использованием демилитаризованной зоны.
9. Сеть на основе многоуровневых доменов.
10. Сеть без использования управляемой маршрутизации.

Экзаменационные вопросы

1. Телекоммуникация, коммуникационная сеть, информационная сеть, вычислительная сеть.
2. Компьютерная сеть (определение, назначение, цель использование).
3. Предпосылки и причины появления сетей.
4. Обобщенная схема функционирования сети.
5. Локальные, корпоративные, региональные и глобальные компьютерные сети.
6. Особенности построения и функционирования, отличия.
7. Конвергенция сетей.
8. Основные принципы построения сети.
9. Многоуровневый подход к решению задачи обмена сообщениями между компьютерами.
10. Основные понятия о протоколе.
11. Стек протоколов.
12. Требования, предъявляемые к современным вычислительным сетям.
13. Проблемные ситуации, возникающие в различных типах сетей, методы и средства их решения.
14. Производительность, надежность и безопасность.
15. Расширяемость и масштабируемость.
16. Прозрачность, управляемость и совместимость.
17. Сети с централизированным управлением, иерархические сети: одноранговые и с выделенным сервером.
18. Технология клиент-сервер. Виды серверов.
19. Понятие топологии при построении компьютерных сетей.
20. Логическая и физическая топологии сети.
21. Топология шина, особенности реализации, коллизия, разделение передающей среды, надежность, безопасность, стоимость реализации.
22. Передающая среда для построения сети по топологии звезда, ограничения, стоимость и безопасность реализации сети. Топологии, в которых отсутствуют коллизии.
23. Особенности реализации топологии кольцо, стоимость и безопасность. С
24. отовая, полносвязная, древовидная и петлевая топологии, как производные топологии, основанные на трех базовых.
25. Области их использования, примеры.
26. Проводная и беспроводная среда передачи. Коаксиальный кабель,

как основная среда для реализации сети по топологии шина.

27. Основные конструктивные элементы, помехозащищенность, технологичность, проблемы обслуживания и монтажа, стоимость.

28. Витая пара, как основная среда для построения сети по топологии звезда.

29. Категории витой пары, отличия, конструктивные элементы, помехозащищенность, ограничения и стоимость реализации.

30. Принцип функционирования оптических сред передачи даны.

31. Одномодовый и многомодовый (с линейным и градиентным коэффициентом преломления) кабель.

32. Радиосети. Радиорелейные сети.

33. Спутниковая связь. Сети транкинговой связи.

34. Инфракрасные беспроводные сети, скорости, расстояния и особенности реализации.

35. Структура, классификация, протоколы систем мобильной связи.

36. Классификация методов доступа к среде передачи.

37. Метод доступа CSMA/CD. Метод доступа CSMA/CA.

38. Метод доступа приоритету. Маркерные методы доступа.

39. Многоуровневая модель OSI, модель и взаимодействие протоколов.

40. Примеры протоколов. Сетевые протоколы. Стеки протоколов.

41. Стандарты локальных сетей. Подуровни канального уровня модели OSI.

42. История появления и характеристика сетей Ethernet. Ограничения и правила построения сетей Ethemet.

43. Расчет времени оборачиваемости сигнала и сокращение межкадрового расстояния.

44. Преодоление ограничений топологий на основе произведенных расчетов.

45. Особенности выбора оборудования и комбинации производных топологий для оптимального функционирования сети.

46. Коммутируемые сети Ethemet. Скоростные версии Ethemet. С

47. Сетевые технологии локальных сетей: ArcNet, Token Ring, FDDI.

48. Гетерогенность и проблемы межсетевого взаимодействия. Основные подходы к организация межсетевого взаимодействия.

49. Мультиплексирование стеков протоколов. Место размещения средств межсетевого взаимодействия.

50. Особенности согласования сетей на транспортном уровне.

51. Источники и типы неоднородностей в транспортной подсистеме.

52. Средства согласования физического уровня. Средства согласования на канальном уровне.

53. Сетевые устройства: повторители, концентраторы, мосты, коммутаторы, маршрутизаторы.

54. Принципы объединения сетей с помощью протоколов сетевого уровня.

55. Семейство протоколов TCP/IP. Транспортные протоколы TCP и UDP.

56. Протокол межсетевого взаимодействия IP, версии протокола. Адресация в IP-сетях.

57. Использование масок и подсетей. Разрешение IP адресов в Ethernet сетях.

58. Маршрутизация IP-адресов. Фрагментация IP-пакетов.

59. Типы протоколов обмена маршрутной информацией.

60. Протоколы DHCP, OSPF, RIP, ARP, RARP.

61. Протокол ICMP.

62. Методы коммутации. Коммутация каналов.

63. Коммутация сообщений. Коммутация пакетов.

64. Мультиплексирование, виды мультиплексирования.

65. Плезиохронная и синхронная цифровые иерархии.

66. Построение компьютерных сетей на основе телефонных сетей с коммутацией каналов.

67. Компьютерные глобальные сети с коммутацией пакетов.

68. Сети X.25, Frame Relay.

69. Технология ATM, основные принципы технологии ATM, стек протоколов ATM, классы сервиса.

70. История возникновения и развития интернета.

71. Принципы построения глобальной компьютерной сети Интернет.

72. Сервисы сети Интернет.

73. Всемирная паутина. URL. Протокол HTTP.

74. Электронная почта. Протоколы электронной почты, почтовые клиенты, безопасность.

75. Протокол передачи файлов.

76. Сетевое управление в IP-сетях.