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

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Б1.Б.14 Сети и телекоммуникации

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

Направление подготовки (специальность) 09.03.01 «Информатика и вычислительная техника»

Профиль образовательной программы «Автоматизированные системы обработки информации и управления**»**

Форма обучения *заочная*

СОДЕРЖАНИЕ

1. КОНСПЕКТ ЛЕКЦИЙ

1.1 Лекция №1 (2 часа).

Тема: «Общие сведения о компьютерных сетях»

1.1.1 Вопросы лекции:

1. Классификация сетей.

2. Топология сетей.

1.1.2 Краткое содержание вопросов:

1. Классификация сетей.

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

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

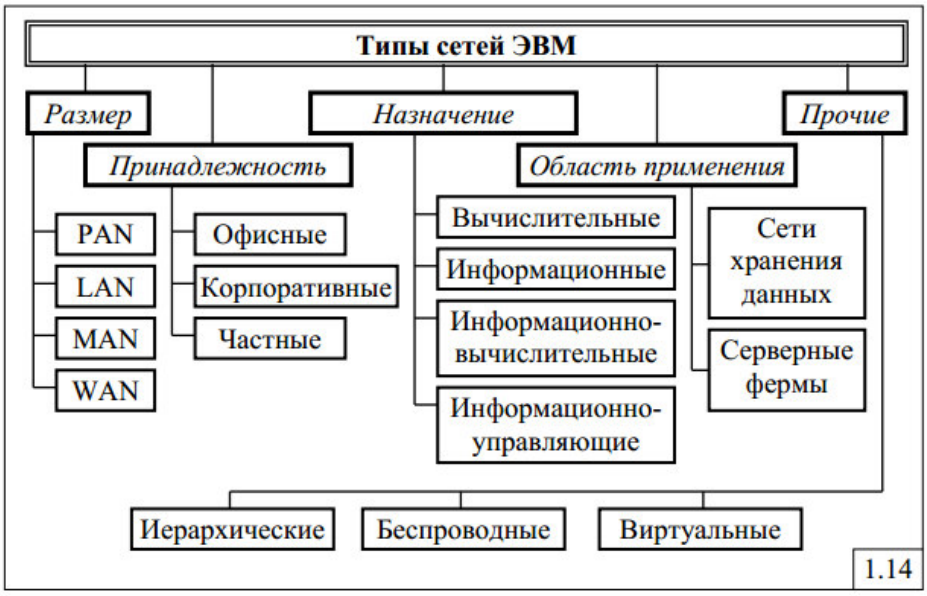


Рисунок 1.Типы сетей ЭВМ

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

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

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

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

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

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

ПримерыЛВС: Ethernet, Token Ring.

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

Протоколы и кабельная система для городской вычислительной сети описываются в стандартах комитета IEEE 802.6. MAN реализуется на основе протокола DQDB (DistributedQueueDualBus) – двойная шина сраспределенной очередью и использует волоконно-оптический кабель для передачи данных со скоростью 100 Мбит/с на территории до 100 км2. MAN может применяться для объединения в одну сеть группы сетей, расположенных в разных зданиях. Последние разработки, связанные с высокоскоростным беспроводным доступом в соответствии со стандартом IEEE 802.16, привели к созданию MAN в виде широкополосных беспроводных ЛВС.

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

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

Настоящей глобальной сетью, пожалуй, можно считать только сеть Интернет. Вряд ли глобальной можно считать сеть, объединяющую 2-3 ЛВС, находящиеся в разных городах, расположенных на расстоянии нескольких десятков или даже сотен километров друг от друга. Однако, поскольку для построения такой «простой» сети используются обычно те же

сетевые технологии и технические средства, что и в сети Интернет, то такие сети обычно тоже относят к классу глобальных сетей.

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

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

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

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

2. Топология сетей.

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

Топология “шина”

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

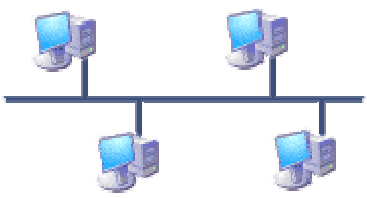


Рисунок 2. Топология шина.

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

Достоинства топологии “шина”:

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

Недостатки топологии “шина”:

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

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

Топология “кольцо”

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

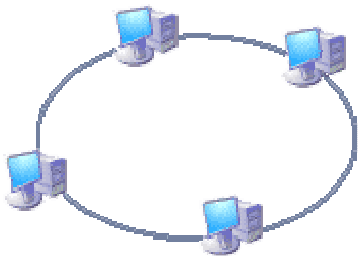


Рисунок 3. Топология кольцо.

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

Достоинства кольцевой топологии:

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

Однако “кольцо” имеет и существенные недостатки:

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

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

Топология “звезда”

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

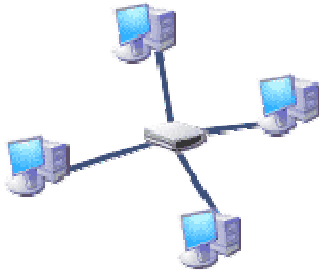


Рисунок 4. Топология звезда.

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

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

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

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

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

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

1.2 Лекция №2 (2 часа).

Тема: «Линии связи».

1.2.1 Вопросы лекции:

1. Спектральный анализ сигналов на линии связи.

2. Характеристики линии связи.

1.2.2 Краткое содержание вопросов:

1. Спектральный анализ сигналов на линии связи.

Из теории гармонического анализа известно, что любой периодический процесс можно представить в виде суммы синусоидальных колебаний различных частот и различных амплитуд (рис. 13). Каждая составляющая синусоида называется также гармоникой, а набор всех гармоник называют спектральным разложением исходного сигнала. Непериодические сигналы можно представить в виде интеграла синусоидальных сигналов с непрерывным спектром частот. Например, спектральное разложение идеального импульса (единичной мощности и нулевой длительности) имеет составляющие всего спектра частот, от - ∞ до + ∞ (рис. 14).

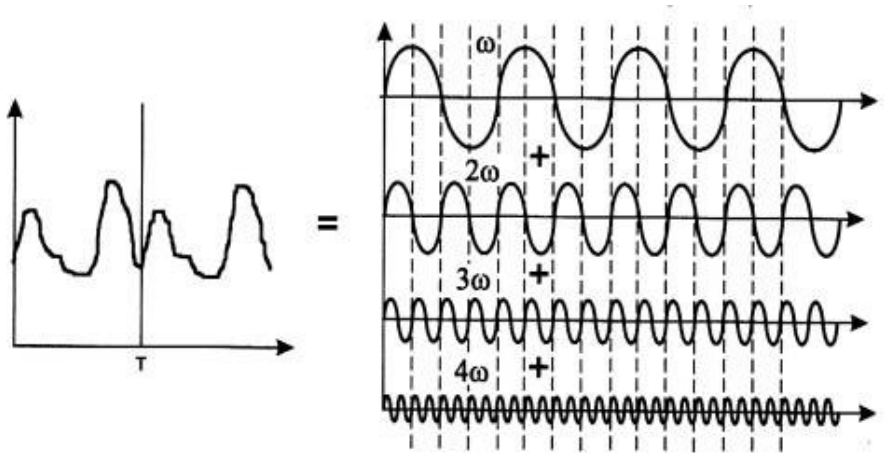


Рисунок 13. Представление периодического сигнала суммой синусоид

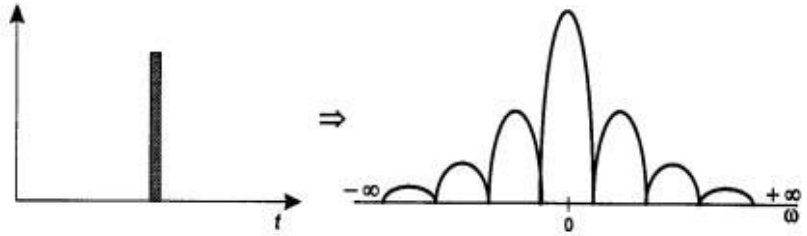


Рисунок 14. Спектральное разложение идеального импульса

Техника нахождения спектра любого исходного сигнала хорошо известна. Для некоторых сигналов, которые хорошо описываются аналитически (например, для последовательности прямоугольных импульсов одинаковой длительности и амплитуды), спектр легко вычисляется на основании формул Фурье. Для сигналов произвольной формы,

встречающихся на практике, спектр можно найти с помощью специальных приборов - спектральных анализаторов, которые измеряют спектр реального сигнала и отображают амплитуды составляющих гармоник на экране или распечатывают их на принтере. Искажение передающим каналом синусоиды какой-либо частоты приводит в конечном счете к искажению передаваемого сигнала любой формы, особенно если синусоиды различных частот искажаются неодинаково. Если это аналоговый сигнал, передающий речь, то изменяется тембр голоса за счет искажения обертонов - боковых частот. При передаче импульсных сигналов, характерных для компьютерных сетей, искажаются низкочастотные и высокочастотные гармоники, в результате фронты импульсов теряют свою прямоугольную форму. Вследствие этого на приемном конце линии сигналы могут плохо распознаваться.

Линия связи искажает передаваемые сигналы из-за того, что ее физические параметры отличаются от идеальных. Так, например, медные провода всегда представляют собой некоторую распределенную по длине комбинацию активного сопротивления, емкостной и индуктивной нагрузки . В результате для синусоид различных частот линия будет обладать различным полным сопротивлением, а значит, и передаваться они будут по-разному. Волоконно-оптический кабель также имеет отклонения, мешающие идеальному распространению света. Если линия связи включает промежуточную аппаратуру, то она также может вносить дополнительные искажения, так как невозможно создать устройства, которые бы одинаково хорошо передавали весь спектр синусоид, от нуля до бесконечности.

Кроме искажений сигналов, вносимых внутренними физическими параметрами линии связи, существуют и внешние помехи, которые вносят свой вклад в искажение формы сигналов на выходе линии. Эти помехи создают различные электрические двигатели, электронные устройства, атмосферные явления и т. д. Несмотря на защитные меры, предпринимаемые разработчиками кабелей и усилительно-коммутирующей аппаратуры, полностью компенсировать влияние внешних помех не удается. Поэтому сигналы на выходе линии связи обычно имеют сложную, по которой иногда трудно понять, какая дискретная информация была подана на вход линии

2. Характеристики линии связи.

К основным характеристикам линий связи относятся:

- амплитудно-частотная характеристика;
- полоса пропускания;
- затухание;
- помехоустойчивость;
- перекрестные наводки на ближнем конце линии;
- пропускная способность;
- достоверность передачи данных;

- удельная стоимость.

В первую очередь разработчика вычислительной сети интересуют пропускная способность и достоверность передачи данных, поскольку эти характеристики прямо влияют на производительность и надежность создаваемой сети. Пропускная способность и достоверность - это характеристики как линии связи, так и способа передачи данных. Поэтому если способ передачи (протокол) уже определен, то известны и эти характеристики. Например, пропускная способность цифровой линии всегда известна, так как на ней определен протокол физического уровня, который задает битовую скорость передачи данных - 64 Кбит/с, 2 Мбит/с и т. п.

Однако нельзя говорить о пропускной способности линии связи, до того как для нее определен протокол физического уровня. Именно в таких случаях, когда только предстоит определить, какой из множества существующих протоколов можно использовать на данной линии, очень важными являются остальные характеристики линии, такие как полоса пропускания, перекрестные наводки, помехоустойчивость и другие характеристики.

Для определения характеристик линии связи часто используют анализ ее реакций на некоторые эталонные воздействия. Такой подход позволяет достаточно просто и однотипно определять характеристики линий связи любой природы, не прибегая к сложным теоретическим исследованиям. Чаще всего в качестве эталонных сигналов для исследования реакций линий связи используются синусоидальные сигналы различных частот. Это связано с тем, что сигналы этого типа часто встречаются в технике и с их помощью можно представить любую функцию времени - как непрерывный процесс колебаний звука, так и прямоугольные импульсы, генерируемые компьютером.

Амплитудно-частотная характеристика (рис. 15) показывает, как затухает амплитуда синусоиды на выходе линии связи по сравнению с амплитудой на ее входе для всех возможных частот передаваемого сигнала. Вместо амплитуды в этой характеристике часто используют также такой параметр сигнала, как его мощность.

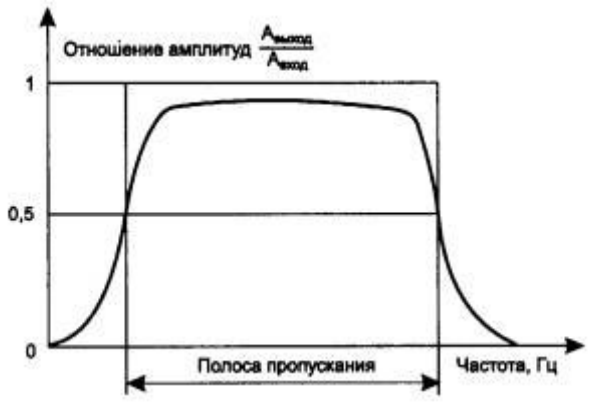


Рисунок 15. Амплитудно-частотная характеристика.

Знание амплитудно-частотной характеристики реальной линии позволяет определить форму выходного сигнала практически для любого входного сигнала. Для этого необходимо найти спектр входного сигнала, преобразовать амплитуду составляющих его гармоник в соответствии с амплитудно-частотной характеристикой, а затем найти форму выходного сигнала, сложив преобразованные гармоники.

Полоса пропускания (bandwidth) – это непрерывный диапазон частот, для которого отношение амплитуды выходного сигнала к входному превышает некоторый заранее заданный предел, обычно 0,5. То есть полоса пропускания определяет диапазон частот синусоидального сигнала, при которых этот сигнал передается по линии связи без значительных искажений. Знание полосы пропускания позволяет получить с некоторой степенью приближения тот же результат, что и знание амплитудно-частотной характеристики. Ширина полосы пропускания в наибольшей степени влияет на максимально возможную скорость передачи информации по линии связи. Именно этот факт нашел отражение в английском эквиваленте рассматриваемого термина (width - ширина).

Затухание (attenuation) определяется как относительное уменьшение амплитуды или мощности сигнала при передаче по линии сигнала определенной частоты. Таким образом, затухание представляет собой одну точку из амплитудно-частотной характеристики линии. Часто при эксплуатации линии заранее известна основная частота передаваемого сигнала, то есть та частота, гармоника которой имеет наибольшую амплитуду и мощность. Поэтому достаточно знать затухание на этой частоте, чтобы приблизительно оценить искажения передаваемых по линии сигналов. Более точные оценки возможны при знании затухания на нескольких частотах, соответствующих нескольким основным гармоникам передаваемого сигнала.

Затухание A обычно измеряется в децибелах (дБ, decibel – dB) и вычисляется по следующей формуле:

$$A = 10 \log_{10} P_{вых} / P_{вх},$$

где $P_{вых}$ – мощность сигнала на выходе линии,

$P_{вх}$ – мощность сигнала на входе линии.

Так как мощность выходного сигнала кабеля без промежуточных усилителей всегда меньше, чем мощность входного сигнала, затухание кабеля всегда является отрицательной величиной.

Абсолютный уровень мощности, например, уровень мощности передатчика, также измеряется в децибелах. При этом в качестве базового значения мощности сигнала, относительно которого измеряется текущая мощность, принимается значение в 1 мВт (милливатт). Таким образом, уровень мощности p вычисляется по следующей формуле:

$$p = 10 \log_{10} P/1мВт \text{ [дБм]},$$

где P - мощность сигнала в милливаттах

дБм (dBm) – это единица измерения уровня мощности (децибел на 1 мВт).

Пропускная способность (throughput) линии характеризует максимально возможную скорость передачи данных по линии связи. Пропускная способность измеряется в битах в секунду – бит/с, а также в производных единицах, таких как килобит в секунду (Кбит/с), мегабит в секунду (Мбит/с), гигабит в секунду (Гбит/с) и т. д.

Пропускная способность линий связи и коммуникационного сетевого оборудования традиционно измеряется в битах в секунду, а не в байтах в секунду.

Пропускная способность линии связи зависит не только от ее характеристик, таких как амплитудно-частотная характеристика, но и от спектра передаваемых сигналов. Если значимые гармоники сигнала (то есть те гармоники, амплитуды которых вносят основной вклад в результирующий сигнал) попадают в полосу пропускания линии, то такой сигнал будет хорошо передаваться данной линией связи и приемник сможет правильно распознать информацию, отправленную по линии передатчиком. Если же значимые гармоники выходят за границы полосы пропускания линии связи, то сигнал будет значительно искажаться, приемник будет ошибаться при распознавании информации, а значит, информация не сможет передаваться с заданной пропускной способностью.

Помехоустойчивость линии определяет ее способность уменьшать уровень помех, создаваемых во внешней среде, на внутренних проводниках. Помехоустойчивость линии зависит от типа используемой физической среды, а также от экранирующих и подавляющих помехи средств самой линии. Наименее помехоустойчивыми являются радиолинии, хорошей устойчивостью обладают кабельные линии и отличной – волоконно-оптические линии, малочувствительные ко внешнему электромагнитному излучению. Обычно для уменьшения помех, появляющихся из-за внешних электромагнитных полей, проводники экранируют и/или скручивают.

Перекрестные наводки на ближнем конце (Near End Cross Talk - NEXT) определяют помехоустойчивость кабеля к внутренним источникам помех, когда электромагнитное поле сигнала, передаваемого выходом передатчика по одной паре проводников, наводит на другую пару проводников сигнал помехи. Если ко второй паре будет подключен приемник, то он может принять наведенную внутреннюю помеху за полезный сигнал. Показатель NEXT, выраженный в децибелах, равен

10 log2 Рвых/Рнав,

где Рвых – мощность выходного сигнала,

Рнав – мощность наведенного сигнала.

Чем меньше значение NEXT, тем лучше кабель. Так, для витой пары категории 5 показатель NEXT должен быть меньше –27 дБ на частоте 100 МГц.

Показатель NEXT обычно используется применительно к кабелю, состоящему из нескольких витых пар, так как в этом случае взаимные наводки одной пары на другую могут достигать значительных величин. Для одинарного коаксиального кабеля (то есть состоящего из одной экранированной жилы) этот показатель не имеет смысла, а для двойного коаксиального кабеля он также не применяется вследствие высокой степени защищенности каждой жилы. Оптические волокна также не создают сколь-нибудь заметных помех друг для друга.

В связи с тем, что в некоторых новых технологиях используется передача данных одновременно по нескольким витым парам, в последнее время стал применяться показатель PowerSUM, являющийся модификацией показателя NEXT. Этот показатель отражает суммарную мощность перекрестных наводок от всех передающих пар в кабеле.

Достоверность передачи данных характеризует вероятность искажения для каждого передаваемого бита данных. Иногда этот же показатель называют интенсивностью битовых ошибок (Bit Error Rate, BER). Величина BER для каналов связи без дополнительных средств защиты от ошибок (например, самокорректирующихся кодов или протоколов с повторной передачей искаженных кадров) составляет, как правило,10-4 – 10-6, в оптоволоконных линиях связи – 10-9. Значение достоверности передачи данных, например, в 10-4 говорит о том, что в среднем из 10000 бит искажается значение одного бита.

Искажения бит происходят как из-за наличия помех на линии, так и по причине искажений формы сигнала ограниченной полосой пропускания линии. Поэтому для повышения достоверности передаваемых данных нужно повышать степень помехозащищенности линии, снижать уровень перекрестных наводок в кабеле, а также использовать более широкополосные линии связи.

1.3 Лекция №3 (2 часа).

Тема: «Кодирование информации»

1.3.1 Вопросы:

1. Выборка способа кодирования.

2. Методы кодирования.

1.3.2 Краткое содержание вопросов:

1. Выборка способа кодирования.

Код — это набор условных обозначений (или сигналов) для записи (или передачи) некоторых заранее определенных понятий.

Кодирование информации – это процесс формирования определенного представления информации. В более узком смысле под термином «кодирование» часто понимают переход

от одной формы представления информации к другой, более удобной для хранения, передачи или обработки.

Обычно каждый образ при кодировании (иногда говорят — шифровке) представлен отдельным знаком.

Знак - это элемент конечного множества отличных друг от друга элементов.

В более узком смысле под термином "кодирование" часто понимают переход от одной формы представления информации к другой, более удобной для хранения, передачи или обработки.

Компьютер может обрабатывать только информацию, представленную в числовой форме. Вся другая информация (например, звуки, изображения, показания приборов и т. д.) для обработки на компьютере должна быть преобразована в числовую форму. Например, чтобы перевести в числовую форму музыкальный звук, можно через небольшие промежутки времени измерять интенсивность звука на определенных частотах, представляя результаты каждого измерения в числовой форме. С помощью программ для компьютера можно выполнить преобразования полученной информации, например "наложить" друг на друга звуки от разных источников.

Аналогичным образом на компьютере можно обрабатывать текстовую информацию. При вводе в компьютер каждая буква кодируется определенным числом, а при выводе на внешние устройства (экран или печать) для восприятия человеком по этим числам строятся изображения букв. Эти изображения называются литерами букв. Соответствие между набором букв и числами называется кодировкой символов.

Как правило, все числа в компьютере представляются с помощью нулей и единиц (а не десяти цифр, как это привычно для людей). Иными словами, компьютеры обычно работают в двоичной системе счисления, поскольку при этом устройства для их обработки получаются значительно более простыми. Ввод чисел в компьютер и вывод их для чтения человеком может осуществляться в привычной десятичной форме, а все необходимые преобразования выполняют программы, работающие на компьютере.

Одна и та же информация может быть представлена (закодирована) в нескольких формах. С появлением компьютеров возникла необходимость кодирования всех видов информации, с которыми имеет дело и отдельный человек, и человечество в целом. Но решать задачу кодирования информации человечество начало задолго до появления компьютеров. Грандиозные достижения человечества - письменность и арифметика - есть не что иное, как система кодирования речи и числовой информации. Информация никогда не появляется в чистом виде, она всегда как-то представлена, как-то закодирована.

Двоичное кодирование – один из распространенных способов представления информации. В вычислительных машинах, в роботах и станках с числовым программным

управлением, как правило, вся информация, с которой имеет дело устройство, кодируется в виде слов двоичного алфавита только двумя знаками 0 и 1 (то есть все информация в памяти компьютера хранится и обрабатывается в виде последовательности нулей и единиц).

Кодирование символьной (текстовой) информации.

Основная операция, производимая над отдельными символами текста - сравнение символов.

При сравнении символов наиболее важными аспектами являются уникальность кода для каждого символа и длина этого кода, а сам выбор принципа кодирования практически не имеет значения.

Для кодирования текстов используются различные таблицы перекодировки. Важно, чтобы при кодировании и декодировании одного и того же текста использовалась одна и та же таблица.

Таблица перекодировки - таблица, содержащая упорядоченный некоторым образом перечень кодируемых символов, в соответствии с которой происходит преобразование символа в его двоичный код и обратно.

Наиболее популярные таблицы перекодировки: ДКОИ-8, ASCII, CP1251, Unicode.

Исторически сложилось, что в качестве длины кода для кодирования символов было выбрано 8 знаков. Любой из знаков 0 или 1 несет в себе 1 бит информации, следовательно один любой символ хранимый в памяти компьютера имеет информационный объем 8 бит (1 байт).

Различных комбинаций из 0 и 1 при длине кода 8 бит может быть 28 = 256, поэтому с помощью одной таблицы перекодировки можно закодировать не более 256 символов. При длине кода в 2 байта (16 бит) можно закодировать 65536 символов.

2. Методы кодирования.

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

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

Распознавание и коррекцию искаженных данных сложно осуществить средствами физического уровня, поэтому чаще всего эту работу берут на себя протоколы, лежащие выше: канальный, сетевой, транспортный или прикладной. С другой стороны, распознавание ошибок на физическом уровне экономит время, так как приемник не ждет полного

помещения кадра в буфер, а отбраковывает его сразу при распознавании ошибочных битов внутри кадра.

Требования, предъявляемые к методам кодирования, являются взаимно противоречивыми, поэтому каждый из рассматриваемых ниже популярных методов цифрового кодирования обладает своими преимуществами и своими недостатками по сравнению с другими.

Потенциальный код без возвращения к нулю отражает то обстоятельство, что при передаче последовательности единиц сигнал не возвращается к нулю в течение такта (как мы увидим ниже, в других методах кодирования возврат к нулю в этом случае происходит). Метод NRZ прост в реализации, обладает хорошей распознаваемостью ошибок (из-за двух резко отличающихся потенциалов), но не обладает свойством самосинхронизации. При передаче длинной последовательности единиц или нулей сигнал на линии не изменяется, поэтому приемник лишен возможности определять по входному сигналу моменты времени, когда нужно в очередной раз считывать данные. Даже при наличии высокоточного тактового генератора приемник может ошибиться с моментом съема данных, так как частоты двух генераторов никогда не бывают полностью идентичными. Поэтому при высоких скоростях обмена данными и длинных последовательностях единиц или нулей небольшое рассогласование тактовых частот может привести к ошибке в целый такт и, соответственно, считыванию некорректного значения бита.

Другим серьезным недостатком метода NRZ является наличие низкочастотной составляющей, которая приближается к нулю при передаче длинных последовательностей единиц или нулей. Из-за этого многие каналы связи, не обеспечивающие прямого гальванического соединения между приемником и источником, этот вид кодирования не поддерживают. В результате в чистом виде код NRZ в сетях не используется. Тем не менее используются его различные модификации, в которых устраняют как плохую самосинхронизацию кода NRZ, так и проблемы постоянной составляющей. Привлекательность кода NRZ, из-за которой имеет смысл заняться его улучшением, состоит в достаточно низкой частоте основной гармоники /o, которая равна N/2 Гц, как это было показано в предыдущем разделе. У других методов кодирования, например манчестерского, основная гармоника имеет более высокую частоту.

Одной из модификаций метода NRZ является метод биполярного кодирования с альтернативной инверсией (Bipolar Alternate Mark Inversion, AMI). В этом методе (рис. 2.16, б) используются три уровня потенциала - отрицательный, нулевой и положительный. Для кодирования логического нуля используется нулевой потенциал, а логическая единица кодируется либо положительным потенциалом, либо отрицательным, при этом потенциал каждой новой единицы противоположен потенциалу предыдущей.

Код AMI частично ликвидирует проблемы постоянной составляющей и отсутствия самосинхронизации, присущие коду NRZ. Это происходит при передаче длинных последовательностей единиц. В этих случаях сигнал на линии представляет собой последовательность разнополярных импульсов с тем же спектром, что и у кода NRZ, передающего чередующиеся нули и единицы, то есть без постоянной составляющей и с основной гармоникой N/2 Гц (где N - битовая скорость передачи данных). Длинные же последовательности нулей также опасны для кода AMI, как и для кода NRZ - сигнал вырождается в постоянный потенциал нулевой амплитуды. Поэтому код AMI требует дальнейшего улучшения, хотя задача упрощается - осталось справиться только с последовательностями нулей.

В целом, для различных комбинаций бит на линии использование кода AMI приводит к более узкому спектру сигнала, чем для кода NRZ, а значит, и к более высокой пропускной способности линии. Например, при передаче чередующихся единиц и нулей основная гармоника fo имеет частоту N/4 Гц. Код AMI предоставляет также некоторые возможности по распознаванию ошибочных сигналов. Так, нарушение строгого чередования полярности сигналов говорит о ложном импульсе или исчезновении с линии корректного импульса. Сигнал с некорректной полярностью называется запрещенным сигналом (signal violation).

В коде AMI используются не два, а три уровня сигнала на линии. Дополнительный уровень требует увеличение мощности передатчика примерно на 3 дБ для обеспечения той же достоверности приема бит на линии, что является общим недостатком кодов с несколькими состояниями сигнала по сравнению с кодами, которые различают только два состояния.

Потенциальный код с инверсией при единице

Существует код, похожий на AMI, но только с двумя уровнями сигнала. При передаче нуля он передает потенциал, который был установлен в предыдущем такте (то есть не меняет его), а при передаче единицы потенциал инвертируется на противоположный. Этот код называется потенциальным кодом с инверсией при единице (Non Return to Zero with ones Inverted, NRZI). Этот код удобен в тех случаях, когда использование третьего уровня сигнала весьма нежелательно, например в оптических кабелях, где устойчиво распознаются два состояния сигнала - свет и темнота.

Для улучшения потенциальных кодов, подобных AMI и NRZI, используются два метода. Первый метод основан на добавлении в исходный код избыточных бит, содержащих логические единицы. Очевидно, что в этом случае длинные последовательности нулей прерываются и код становится самосинхронизирующимся для любых передаваемых данных. Исчезает также постоянная составляющая, а значит, еще более сужается спектр сигнала. Но этот метод снижает полезную пропускную способность линии, так как избыточные единицы

пользовательской информации не несут. Другой метод основан на предварительном «перемешивании» исходной информации таким образом, чтобы вероятность появления единиц и нулей на линии становилась близкой. Устройства, или блоки, выполняющие такую операцию, называются трамблерами (scramble - свалка, беспорядочная сборка). При скремблировании используется известный алгоритм, поэтому приемник, получив двоичные данные, передает их на дескрэмблер, который восстанавливает исходную последовательность бит. Избыточные биты при этом по линии не передаются. Оба метода относятся к логическому, а не физическому кодированию, так как форму сигналов на линии они не определяют. Более детально они изучаются в следующем разделе.

Биполярный импульсный код

Кроме потенциальных кодов в сетях используются и импульсные коды, когда данные представлены полным импульсом или же его частью - фронтом. Наиболее простым случаем такого подхода является биполярный импульсный код, в котором единица представлена импульсом одной полярности, а ноль - другой. Каждый импульс длится половину такта. Такой код обладает отличными самосинхронизирующими свойствами, но постоянная составляющая, может присутствовать, например, при передаче длинной последовательности единиц или нулей. Кроме того, спектр у него шире, чем у потенциальных кодов. Так, при передаче всех нулей или единиц частота основной гармоники кода будет равна N Гц, что в два раза выше основной гармоники кода NRZ и в четыре раза выше основной гармоники кода AMI при передаче чередующихся единиц и нулей. Из-за слишком широкого спектра биполярный импульсный код используется редко.

Манчестерский код

В локальных сетях до недавнего времени самым распространенным методом кодирования был так называемый манчестерский код. Он применяется в технологиях Ethernet и Token Ring.

В манчестерском коде для кодирования единиц и нулей используется перепад потенциала, то есть фронт импульса. При манчестерском кодировании каждый такт делится на две части. Информация кодируется перепадами потенциала, происходящими в середине каждого такта. Единица кодируется перепадом от низкого уровня сигнала к высокому, а ноль - обратным перепадом. В начале каждого такта может происходить служебный перепад сигнала, если нужно представить несколько единиц или нулей подряд. Так как сигнал изменяется по крайней мере один раз за такт передачи одного бита данных, то манчестерский код обладает хорошими самосинхронизирующими свойствами. Полоса пропускания манчестерского кода уже, чем у биполярного импульсного. У него также нет постоянной составляющей, а основная гармоника в худшем случае (при передаче последовательности единиц или нулей) имеет частоту N Гц, а в лучшем (при передаче чередующихся единиц и

нулей) она равна N/2 Гц, как и у кодов AMI или NRZ. В среднем ширина полосы манчестерского кода в полтора раза уже, чем у биполярного импульсного кода, а основная гармоника колеблется вблизи значения 3N/4. Манчестерский код имеет еще одно преимущество перед биполярным импульсным кодом. В последнем для передачи данных используются три уровня сигнала, а в манчестерском - два.

Потенциальный код 2B1Q

Потенциальный код с четырьмя уровнями сигнала для кодирования данных. Это код 2B1Q, название которого отражает его суть - каждые два бита (2B) передаются за один такт сигналом, имеющим четыре состояния (1Q), Паре бит 00 соответствует потенциал -2,5 В, паре бит 01 соответствует потенциал -0,833 В, паре 11 - потенциал +0,833 В, а паре 10 - потенциал +2,5 В. При этом способе кодирования требуются дополнительные меры по борьбе с длинными последовательностями одинаковых пар бит, так как при этом сигнал превращается в постоянную составляющую. При случайном чередовании бит спектр сигнала в два раза уже, чем у кода NRZ, так как при той же битовой скорости длительность такта увеличивается в два раза. Таким образом, с помощью кода 2B1Q можно по одной и той же линии передавать данные в два раза быстрее, чем с помощью кода AMI или NRZI. Однако для его реализации мощность передатчика должна быть выше, чтобы четыре уровня четко различались приемником на фоне помех.

2. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

2.1 Лабораторная работа № 1 (2 часа)

Тема: «Общие сведения о компьютерных сетях»

2.1.1 Цель работы: ознакомится с вычислительными сетями, классификацией сетей. Приобрести навыки проектирования локальной вычислительной сети для конкретной организации с использованием различного типа оборудования. Научиться работать с локальными вычислительными сетям, кабельной системой, оборудованием (серверами, концентраторами, сетевыми адаптерами), использовать различные топологии локальных сетей.

2.1.2 Задачи работы:

1. Спроектировать ЛВС для организации, располагающейся в здании, состоящего из различного количества этажей и комнат на этажах;

2. При проектировании учитывать различные типы топологий (шина, звезда, кольцо);

3. Использовать различные типы сред передачи данных: сетевые кабели (коаскиальный кабель, витая пара проводов, оптоволокно), провода, радиоканалы наземной и спутниковой связи и т.д.

2.1.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. Компьютерная ЛВС.

2.1.4 Описание (ход) работы:

*Вычислительной сетью*называется система, состоящая из двух или более удаленных ЭВМ, соединенных с помощью специальной аппаратуры и взаимодействующих между собой по каналам передачи данных.

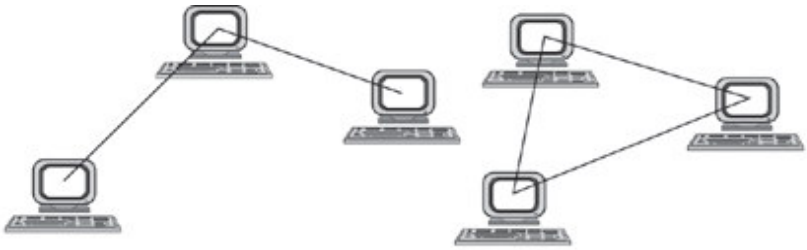
Самая простая сеть (network) состоит из нескольких персональных компьютеров, соединенных между собой сетевым кабелем. При этом в каждом компьютере устанавливается специальная плата сетевого адаптера (NIC), осуществляющая связь между системной шиной компьютера и сетевым кабелем.

Кроме этого, все компьютерные сети работают под управлением специальной сетевой операционной системы (NOS – Network Operation System). Основное назначение компьютерных сетей – совместное использование ресурсов и осуществление интерактивной связи как внутри одной фирмы, так и за ее пределами.

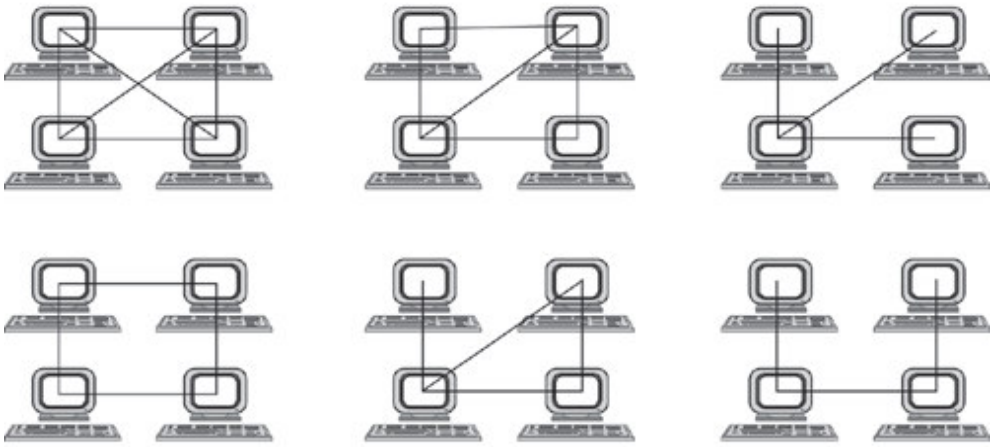
Как только компьютеров становится больше двух, возникает проблема выбора **конфигурации физических связей** или **топологии**. Под топологией сети понимается конфигурация графа, вершинам которого соответствуют конечные узлы сети (например, компьютеры) и коммуникационное оборудование (например, маршрутизаторы),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

Число возможных конфигураций резко возрастает при увеличении числа связываемых устройств. Так, если три компьютера мы можем связать двумя способами, то для четырех компьютеров (рисунок 1) можно предложить уже шесть топологически различных конфигураций (при условии неразличимости компьютеров).

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



а) вариант связи трех компьютеров



б) вариант связи четырех компьютеров

Рисунок 1 - Варианты связи компьютеров

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

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

Полносвязная топология (рисунок 2) соответствует сети, в которой каждый компьютер непосредственно связан со всеми остальными. Несмотря на логическую простоту, это вариант громоздкий и неэффективный. Действительно, каждый компьютер в сети должен

иметь большое количество коммуникационных портов, достаточное для связи с каждым из остальных компьютеров. Для каждой пары компьютеров должна быть выделена отдельная физическая линия связи (в некоторых случаях даже две, если невозможно использование этой линии для двусторонней передачи.) Полносвязные топологии в крупных сетях применяются редко, так как для связи N узлов требуется N(N-1)/2 физических дуплексных линий связи, т.е. имеет место квадратическая зависимость. Чаще этот вид топологии используется в многомашинных комплексах или в сетях, объединяющих небольшое количество компьютеров.

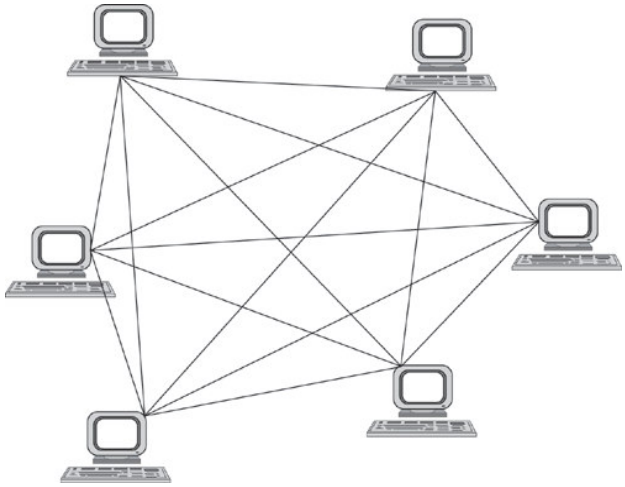


Рисунок 2 - Полносвязная конфигурация

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

Ячеистая топология получается из полносвязной путем удаления некоторых возможных связей. Ячеистая топология допускает соединение большого количества компьютеров и характерна для крупных сетей (рисунок 3).

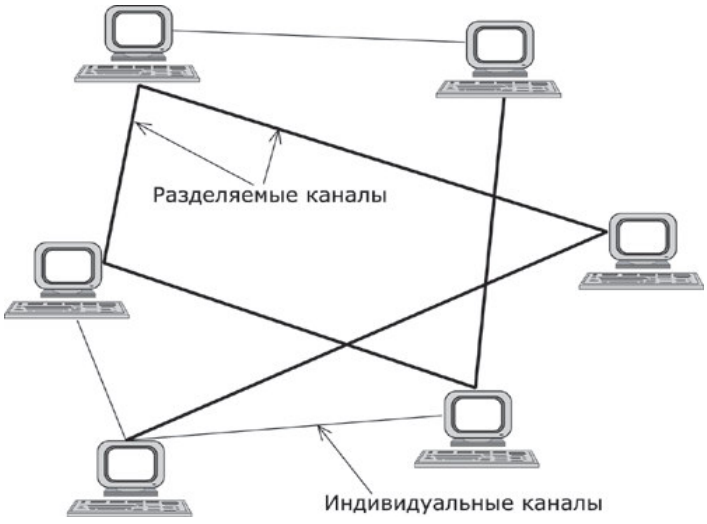


Рисунок 3 - Ячеистая топология

В сетях с кольцевой конфигурацией (рисунок 4) данные передаются по кольцу от одного компьютера к другому. Главное достоинство "кольца" в том, что оно по своей природе обладает свойством резервирования связей.

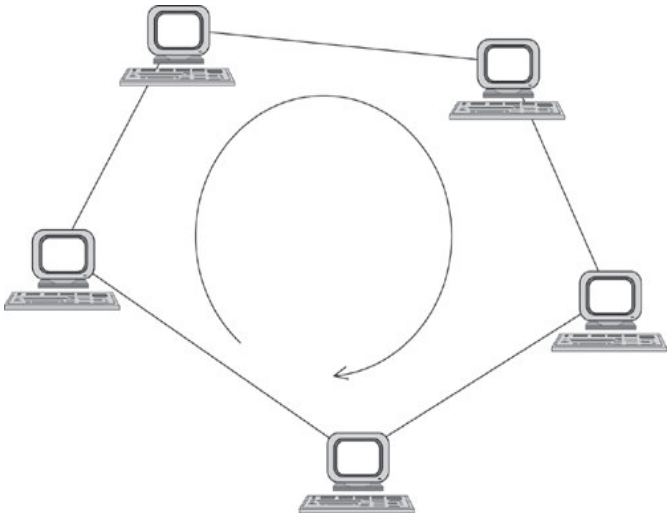


Рисунок 4 -Топология "кольцо"

Действительно, любая пара узлов соединена здесь двумя путями — по часовой стрелке и против. "Кольцо" представляет собой очень удобную конфигурацию и для организации обратной связи — данные, сделав полный оборот, возвращаются к узлу-источнику. Поэтому отправитель в данном случае может контролировать процесс доставки данных адресату. Часто это свойство "кольца" используется для тестирования связности сети и поиска узла, работающего некорректно. В то же время в сетях с кольцевой топологией необходимо принимать специальные меры, чтобы в случае выхода из строя или отключения какой-либо станции не прерывался канал связи между остальными станциями "кольца".

Топология "звезда" (рисунок 5) образуется в том случае, когда каждый компьютер с помощью отдельного кабеля подключается к общему центральному устройству, называемому концентратором.

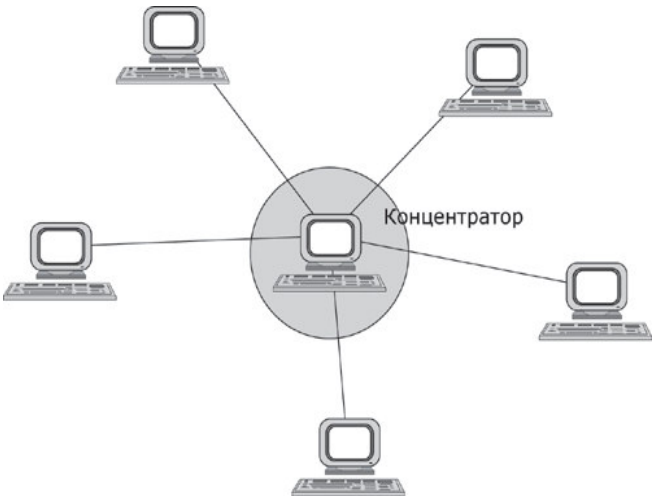


Рисунок 5 - Топология "звезда"

В функции концентратора входит направление передаваемой компьютером информации одному или всем остальным компьютерам сети. В роли концентратора может выступать как компьютер, так и специализированное устройство, такое как многопортовый повторитель, коммутатор или маршрутизатор. К недостаткам топологии типа "звезда" относится более высокая стоимость сетевого оборудования, связанная с необходимостью приобретения специализированного центрального устройства. Кроме того, возможности наращивания количества узлов в сети ограничиваются количеством портов концентратора.

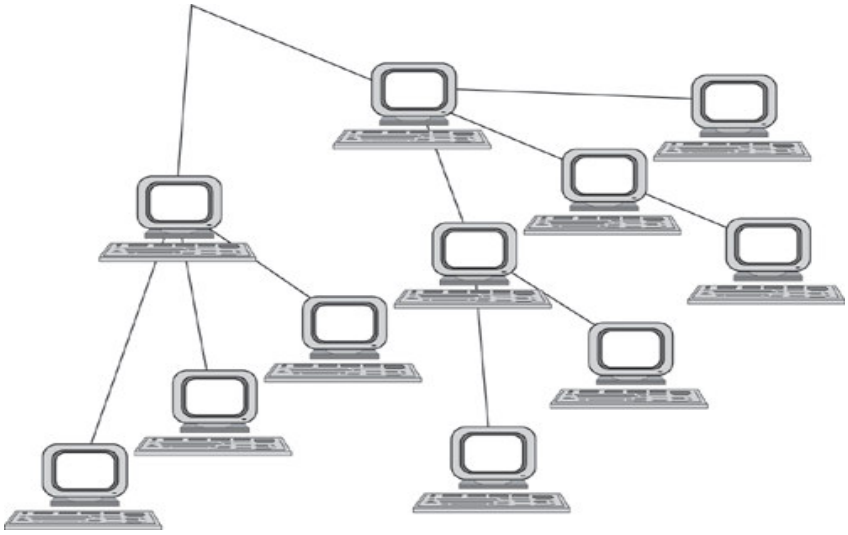


Рисунок 6 - Топология "иерархическая звезда" или "дерево"

Иногда имеет смысл строить сеть с использованием нескольких концентраторов, иерархически соединенных между собой связями типа "звезда" (рисунок 6).

Получаемую в результате **структуру** называют также деревом. В настоящее время дерево является самым распространенным типом топологии связей, как в локальных, так и в глобальных сетях.

Особым частным случаем конфигурации, звезда является конфигурация "общая шина" (рисунок 7).

Здесь в роли центрального элемента выступает пассивный кабель, к которому по схеме "монтажного ИЛИ" подключается несколько компьютеров (такую же топологию имеют многие сети, использующие беспроводную связь — роль общей шины здесь играет общая радиосреда).

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

Рисунок 7 - Топология "общая шина"

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

Самым серьезным недостатком "общей шины" является ее недостаточная надежность: любой дефект кабеля или какого-нибудь из многочисленных разъемов полностью парализует всю сеть.

Другой недостаток "общей шины" — невысокая производительность, так как при таком способе подключения в каждый момент времени только один компьютер может передавать данные по сети, поэтому пропускная способность канала связи всегда делится между всеми узлами сети. До недавнего времени "общая шина" являлась одной из самых популярных топологий для локальных сетей.

В то время как небольшие сети, как правило, имеют типовую топологию — **"звезда"**, **"кольцо"** или **"общая шина"**, для крупных сетей характерно наличие произвольных связей между компьютерами.

В таких сетях можно выделить отдельные произвольно связанные фрагменты (подсети), имеющие типовую топологию, поэтому их называют сетями со **смешанной** топологией (рисунок 8).

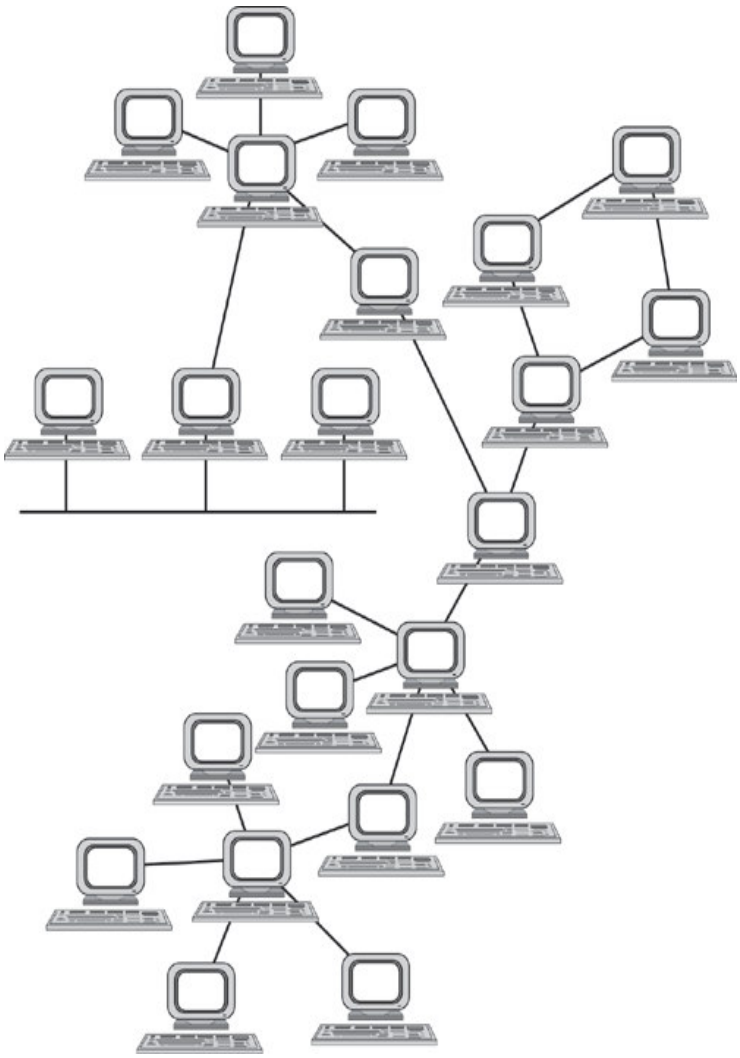


Рисунок 8 - Смешанная топология

Линия связи (рисунок 9) состоит в общем случае из физической среды, по которой передаются электрические информационные сигналы, аппаратуры передачи данных и промежуточной аппаратуры. Синонимом термина *линия связи (line)* является термин *канал связи (channel)*.

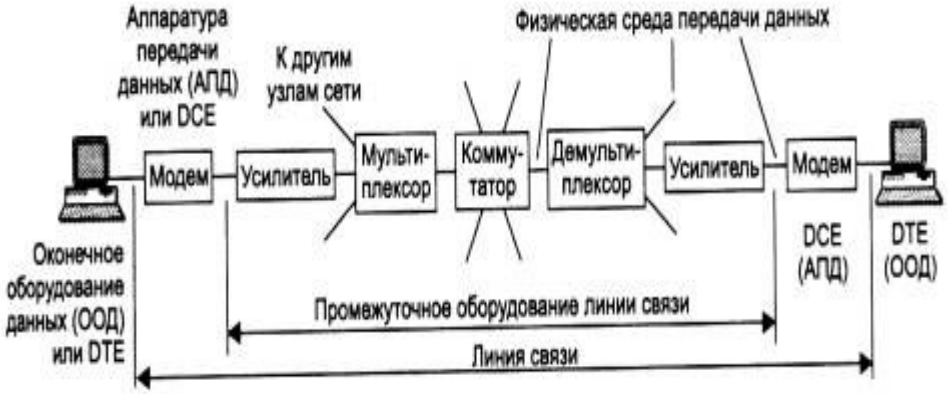


Рисунок 9 – Линии связи

Физическая среда передачи данных (medium) может представлять собой кабель, то есть набор проводов, изоляционных и защитных оболочек и соединительных разъемов, а также земную атмосферу или космическое пространство, через которые распространяются электромагнитные волны.

В зависимости от среды передачи данных линии связи разделяются на следующие: проводные (воздушные), кабельные (медные и волоконно-оптические), радиоканалы наземной и спутниковой связи (рисунок 10):

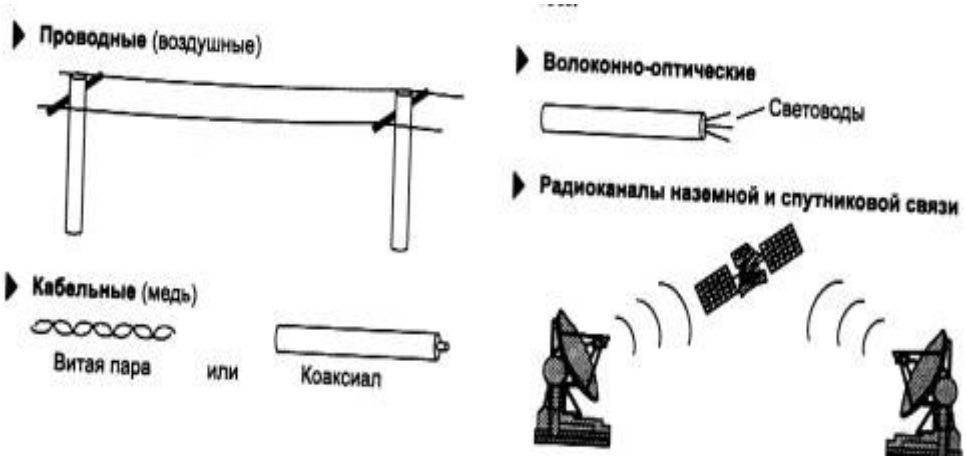


Рисунок 10 – Типы линий связи

Проводные (воздушные) линии связи представляют собой провода без каких-либо изолирующих или экранирующих оплеток, проложенные между столбами и висящие в воздухе. По таким линиям связи традиционно передаются телефонные или телеграфные сигналы, но при отсутствии других возможностей эти линии используются и для передачи компьютерных данных. Скоростные качества и помехозащищенность этих линий оставляют желать много лучшего. Сегодня проводные линии связи быстро вытесняются кабельными.

Кабельные линии представляют собой достаточно сложную конструкцию. Кабель состоит из проводников, заключенных в несколько слоев изоляции: электрической, электромагнитной, механической, а также, возможно, климатической. Кроме того, кабель может быть оснащен разъемами, позволяющими быстро выполнять присоединение к нему различного оборудования. В компьютерных сетях применяются три основных типа кабеля: кабели на основе скрученных пар медных проводов, коаксиальные кабели с медной жилой, а также волоконно-оптические кабели.

Скрученная пара проводов называется *витой парой (twisted pair)*. Витая пара существует в экранированном варианте *(Shielded Twistedpair, STP),* когда пара медных проводов обертывается в изоляционный экран, и неэкранированном *(Unshielded Twistedpair, UTP)*, когда изоляционная обертка отсутствует. Скручивание проводов снижает влияние внешних помех на полезные сигналы, передаваемые по кабелю.

Коаксиальный кабель (coaxial) имеет несимметричную конструкцию и состоит из внутренней медной жилы и оплетки, отделенной от жилы слоем изоляции. Существует несколько типов коаксиального кабеля, отличающихся характеристиками и областями применения - для локальных сетей, для глобальных сетей, для кабельного телевидения и т. п.

Волоконно-оптический кабель (optical fiber) состоит из тонких (5-60 микрон) волокон, по которым распространяются световые сигналы. Это наиболее качественный тип кабеля - он обеспечивает передачу данных с очень высокой скоростью (до 10 Гбит/с и выше) и к тому же лучше других типов передающей среды обеспечивает защиту данных от внешних помех.

Радиоканалы наземной и спутниковой связи образуются с помощью передатчика и приемника радиоволн. Существует большое количество различных типов радиоканалов, отличающихся как используемым частотным диапазоном, так и дальностью канала (КВ, СВ и ДВ, УКВ, СВЧ или microwaves).

В компьютерных сетях сегодня применяются практически все описанные типы физических сред передачи данных, но наиболее перспективными являются волоконно-оптические. Популярной средой является также витая пара. С помощью витой пары обычно подключают конечных абонентов сетей на расстояниях до 100 метров от концентратора. Спутниковые каналы и радиосвязь используются чаще всего в тех случаях, когда кабельные связи применить нельзя - например, при прохождении канала через малонаселенную местность или же для связи с мобильным пользователем сети. В таблице 1 приведены основные отличия разных типов кабелей.

Таблица 1 - Сетевые кабели

Характеристика	Тонкий коак-сиальный кабель	Толстый коаксиальный кабель	Витая пара	Оптоволоконный кабель
Эффективная длина кабеля	185 м	500м	100м	2км
Скорость передачи	10 Мбит/с	10 Мбит/с	$\geq$ 10 Мбит/с	$\geq$ 10 Мбит/с
Гибкость	Довольно гибкий	Менее гибкий	Самый гибкий	Не гибкий
Подверженность помехам	Хорошо защищен	Хорошо Защищен	Подвержен помехам	Не подвержен помехам

Варианты заданий на лабораторную работу

Проектируемая локальная вычислительная сеть должна быть для организации, располагающейся в нескольких зданиях (варианты представлены в таблице А.1).

Таблица А.1

Вариант	Количество зданий	Расстояние между зданиями, м	Количество этажей в зданиях	Число комнат на каждом этаже	Общее число компьютеров	Число используемых концентраторов
1	2	200	2	6	60	6
2	3	340	5	3	60	7
3	2	820	3	5	60	7
4	2	250	4	6	72	8
5	3	1420	2	5	50	5
6	1	230	5	2	50	5
7	2	200	4	2	40	4
8	2	720	4	3	60	8
9	3	250	3	6	54	5
10	4	1320	2	7	70	7
11	2	450	6	4	72	6
12	1	250	5	4	60	6
13	2	500	4	4	64	7
14	3	650	3	4	60	8
15	4	1250	2	4	40	4
16	3	400	6	3	54	5
17	2	200	7	2	70	7
18	1	500	1	3	45	4
19	2	120	7	3	63	7
20	3	430	6	2	60	6

Необходимо добиться максимальной эффективности использования сети по критерию цена-качество-скорость.

Составить схему ЛВС.

Подготовить отчет.

2.2 Лабораторная работа № 2 (2 часа)

Тема: «Коммутация»

2.2.1 Цель работы: изучить таблицу коммутации и Web-интерфейс коммутатора D-Link.

2.2.2 Задачи работы:

1. Изучить таблицу коммутации;

2. Изучить Web-интерфейс коммутатора D-Link.

2.2.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. Рабочая станция;

2. Коммутатор DES-3200-10;

3. Кабель Ethernet;

4. Консольный кабель.

2.2.4 Описание (ход) работы:

Коммутатор (switch) — основное активное сетевое оборудование современныхлокальных сетей. В отличие от концентратора, коммутатор работает на канальном уровне модели OSI и передает кадры не на все порты, а непосредственно получателю, анализируя MAC-адрес источника/назначения.

Передача кадров коммутатором осуществляется на основе *таблицы коммутации*. Каждая запись в таблице коммутации состоит из номера порта и MAC-адреса. Как создаются записи в таблице коммутации? Например, если на порт 1 коммутатора поступает кадр от рабочей станции ПК1, то в таблице создается запись, ассоциирующая MAC-адрес рабочей станции ПК1 с номером входного порта. Таблица коммутации может строиться коммутатором автоматически, на основе динамического изучения MAC-адресов источников поступающих на порты кадров, или создаваться вручную администратором сети.

Коммутируемые сети имеют ряд особенностей и ограничений. Одной из главных проблем таких сетей, является увеличение широковещательных доменов. *Широковещательный домен* – это область распространения широковещательного трафика.Широковещательные кадры передаются на все узлы сети и могут привести к нерациональному использованию полосы пропускания. Для того, чтобы этого не происходило, нужно организовать небольшие широковещательные домены или *виртуальныелокальные сети (VLAN)*.

Виртуальной локальной сетью называется логическая группа узлов сети, трафиккоторой, в том числе и широковещательный, на канальном уровне полностью изолирован от других узлов сети. Это значит, что передача кадров между разными виртуальными локальными сетями на основе MAC-адреса невозможна независимо от типа адреса — уникального, группового или широковещательного. В то же время, внутри

виртуальной локальной сети кадры передаются по технологии коммутации, то есть только на тот порт, который связан с MAC-адресом назначения кадра.

Управление коммутатором через Web-интерфейс и изучение таблицы коммутации

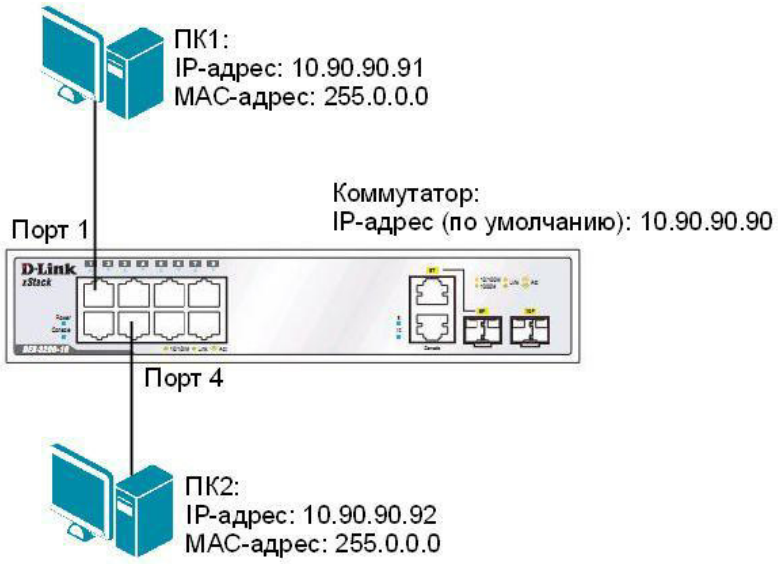


Рисунок 1.1 Схема сети

Шаг 1. Подключите ПК1 и ПК2 к коммутатору как показано на рис. 1.1.

Шаг 2. Настройте на рабочей станции ПК1 и ПК2 статический IP-адрес.

Шаг 3. Проверьте доступность соединения между рабочими станциями ПК1 и ПК2.

В командной строке ПК1 введите: ping 10.90.90.92

В командной строке ПК2 введите: ping 10.90.90.91

Шаг 4. Зайдите на Web-интерфейс коммутатора.

Чтобы зайти на Web -интерфейс коммутатора, выполните следующие действия:

1. На рабочей станции ПК1 запустите Web-браузер (Internet Explorer, Mozilla Firefox), в адресной строке которого укажите IP-адрес интерфейса управления коммутатора по умолчанию:

http://10.90.90.90

Внимание: IP-адрес управления коммутатора по умолчанию обычно указывается в руководствепользователя. Для коммутатора D-Link DES-3200-10 IP-адрес управления по умолчанию — 10.90.90.90

2. В появившемся окне аутентификации, поля *User name* и *Password* оставьте пустыми и нажмите *Ок*. После этого появится окно Web-интерфейса управления коммутатора (рис. 1.2). Если на рабочей станции произведены настройки прокси-сервера, то их нужно отключить.

Для Mozilla Firefox: меню *Инструменты → Настройки → Дополнительные*. Далее вкладка *Сеть →Настройка параметров соединения Firefox с Интернетом → Настроить → Без прокси*.

Для Internet Explorer: меню *Сервис → Свойство обозревателя*. Далее вкладка *Подключения →Настройка сети → Автоматическое определение параметров*.

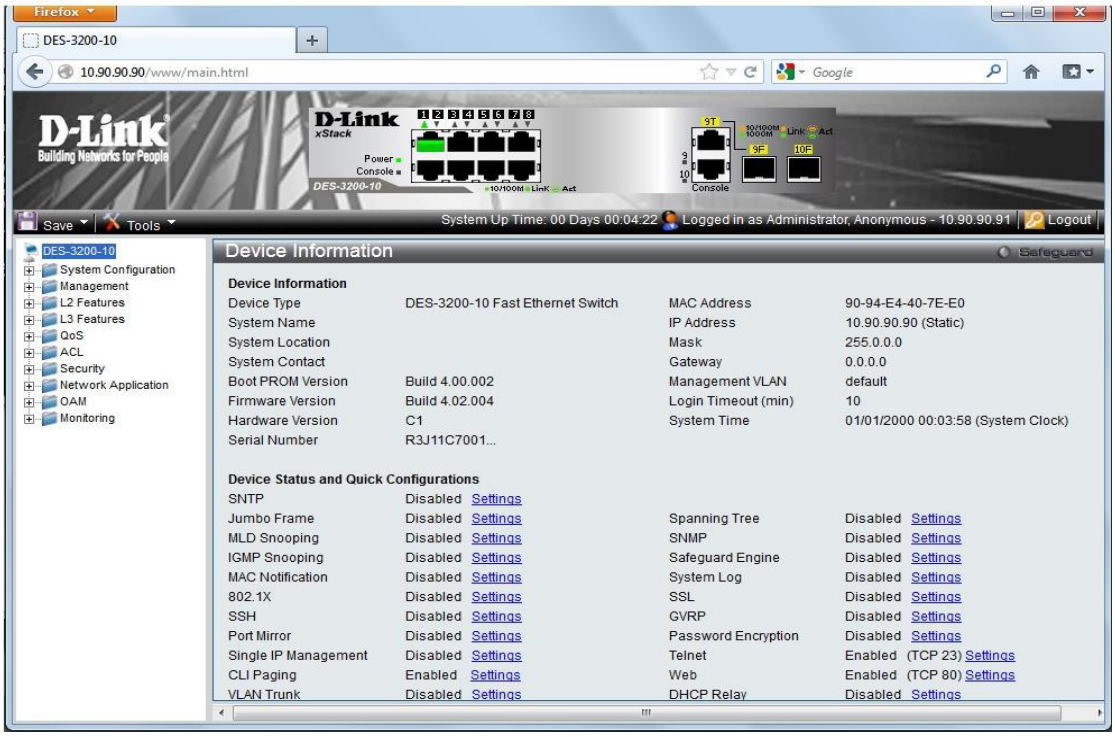


Рисунок 1.2 Web-интерфейс управления коммутатора DES-3200-10

Шаг 5. Посмотрите содержимое таблицы коммутации. В левой части окна выберите*L2Features → FDB → MAC Address Table* (рис. 1.3).

Сколько записей наблюдаете? _____

Какой тип (type) у каждой записи в таблице коммутации? _____

Шаг 6. Отключите рабочую станцию ПК2 от 4 порта и подключите к 5 порту.

Шаг 7. Посмотрите содержимое таблицы коммутации. Что изменилось? _____

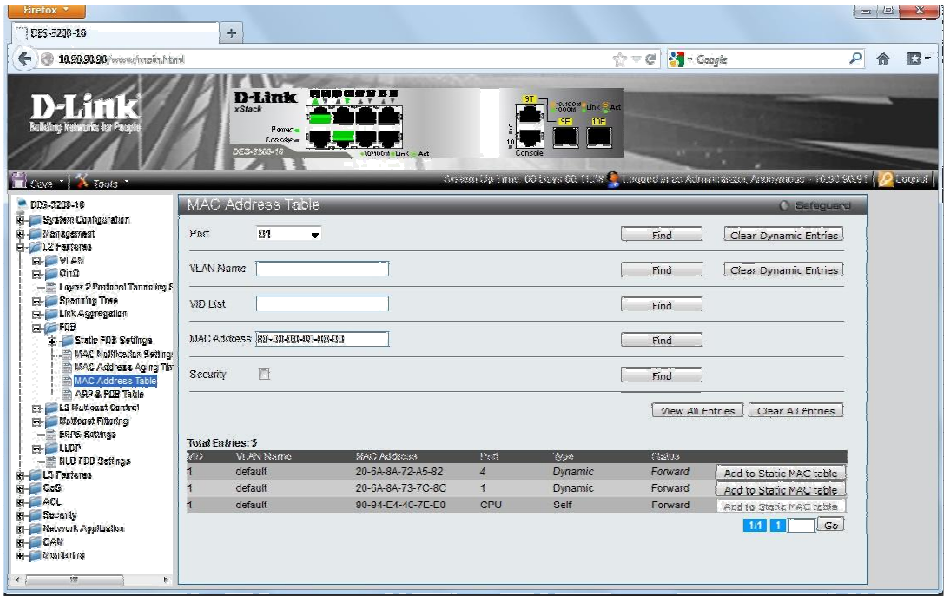


Рисунок 1.3 Таблица коммутации

Шаг 8. Создайте статическую запись в таблице коммутации для ПК2 на порте 5. Для этогонажмите на кнопку *Add to Static MAC table* (рис. 1.4).

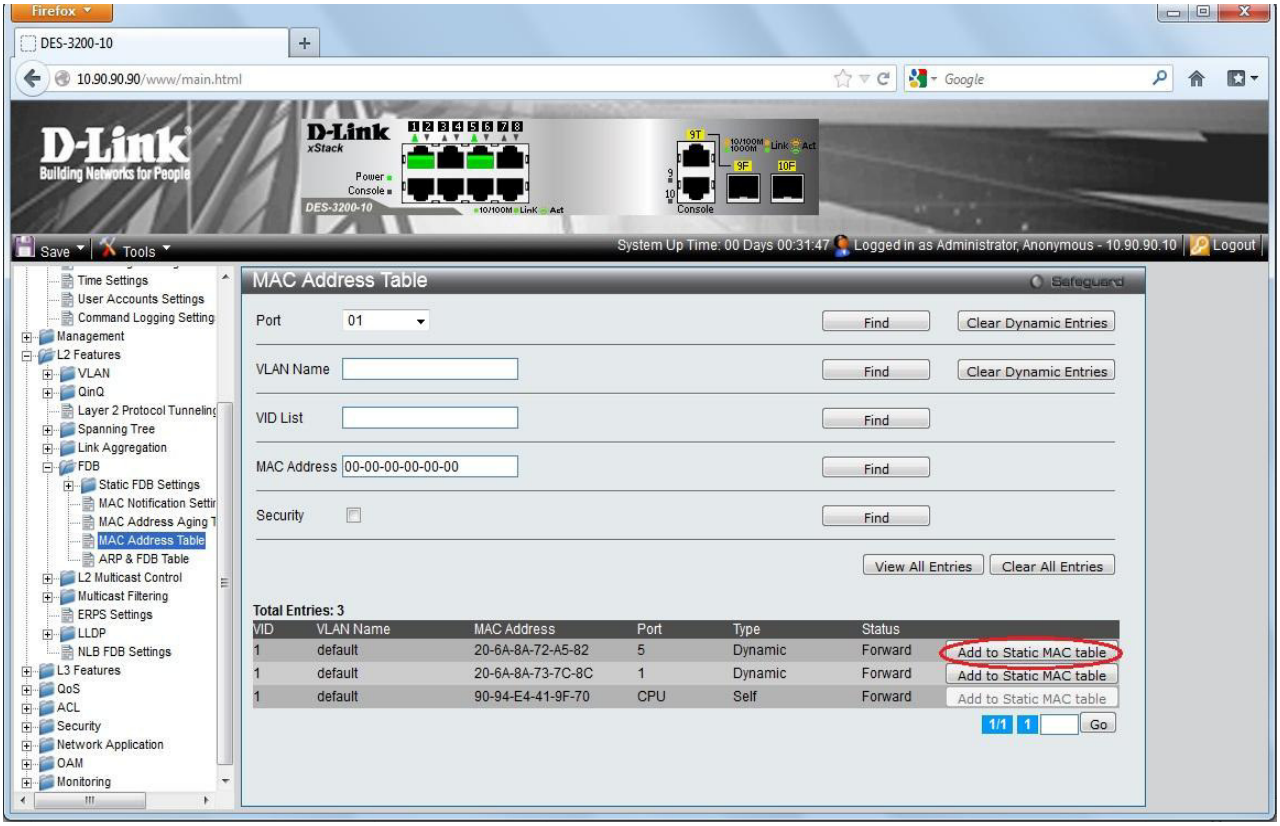


Рисунок 1.4 Создание статической записи

Шаг 9. Отключите рабочую станцию ПК2 от 5 порта и подключите к 4 порту.

Шаг 10. Проверьте доступность соединения между рабочими станциями ПК1 и ПК2.

В командной строке ПК1введите: ping 10.90.90.92

В командной строке ПК2 введите: ping 10.90.90.91

Объясните, почему нет связи между ПК1 и ПК2 _____

Шаг 11. Удалите статическую запись из таблицы коммутации. Влевойчастиокнавыберите *FDB → Static FDB Settings → Unicast Static FDB Settings*.В правой части окна нажмите *Delete* напротив записи для ПК2 (рис. 1.5).

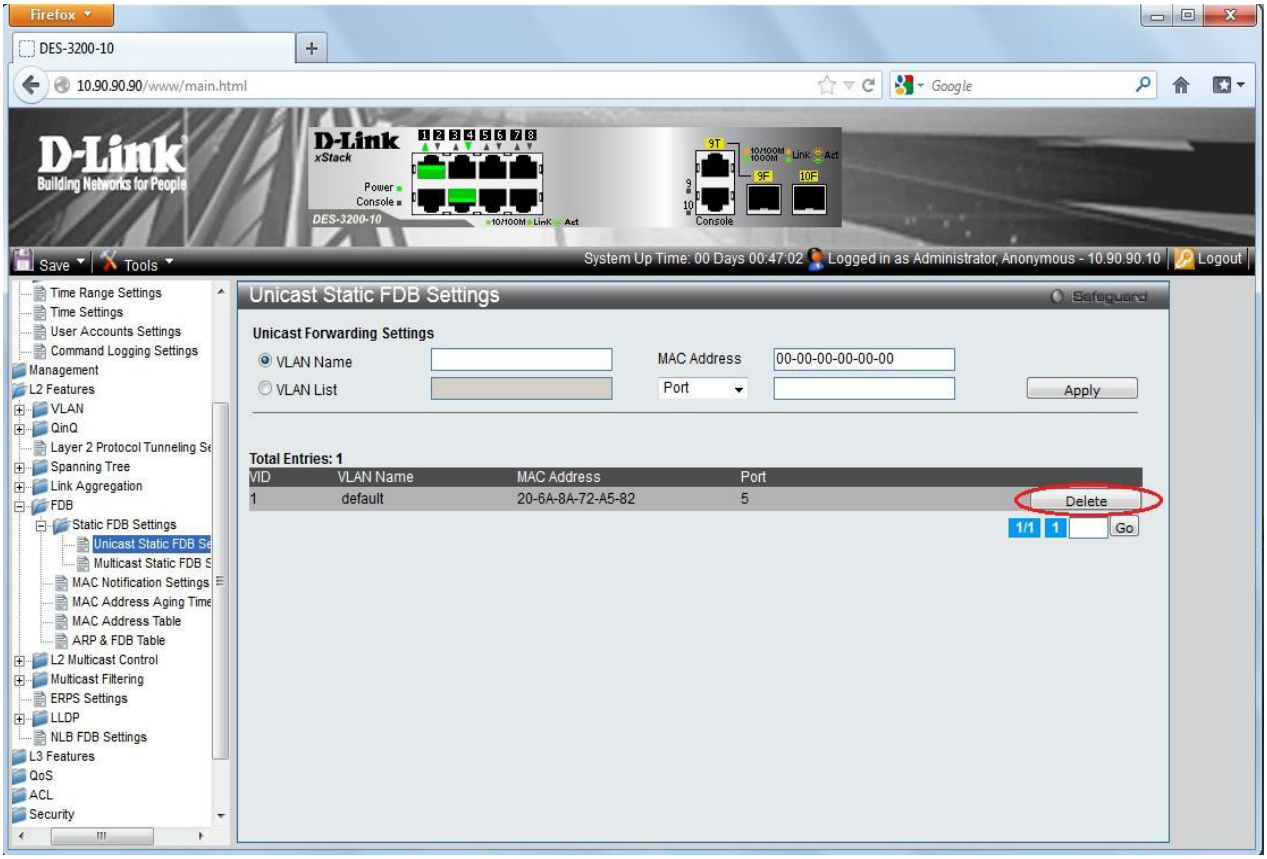


Рисунок 1.5 Удаление статической записи

Шаг 12. Проверьте доступность соединения между рабочими станциями ПК1 и ПК2.

В командной строке ПК1введите: ping 10.90.90.92

В командной строке ПК2 введите: ping 10.90.90.91

Шаг 13. Сбросьте настройки коммутатора к заводским настройкам по умолчанию. Выберите *Tools* → *Reset* → *ResetConfig*инажмите*Apply*.

Диагностика сети во время широковещательного шторма

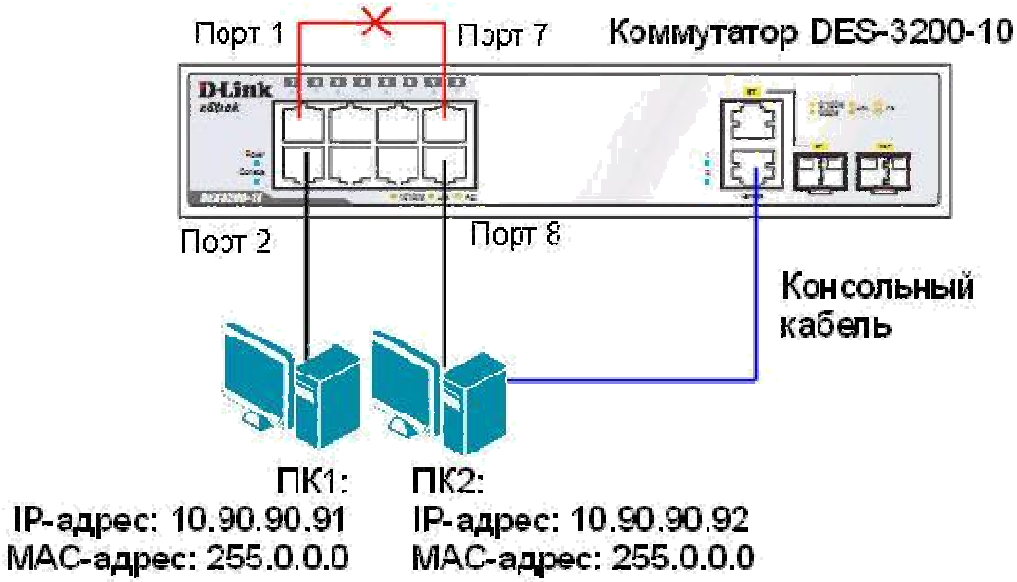


Рисунок 1.6 Схема сети

Управление коммутатором осуществляется не только через Web-интерфейс. Для более тонкой настройки устройства используется управление через интерфейс командной строки (Command Line Interface, CLI). Доступ к интерфейсу командной строки коммутатора осуществляется путем подключения к его консольному порту персонального компьютера с установленной программой эмуляции терминала.

Шаг 1. Подключите ПК2 к консольному порту коммутатора с помощью кабеля RS-232.После подключения к консольному порту коммутатора, на персональном компьютере запустите программу эмуляции терминала VT100 (например, *putty.exe* или программу *HyperTerminal* в Windows XP).

В программе HyperTerminal установите следующие параметры подключения:

Скорость (бит/с):	115200
Биты данных:	8
Чётность:	нет
Стоповые биты:	1
Управление потоком:	нет

В программе *Putty* установите следующие параметры подключения:

1. В категории *Session* выберите *Serial* и установите скорость 115200 (рис. 1.7);

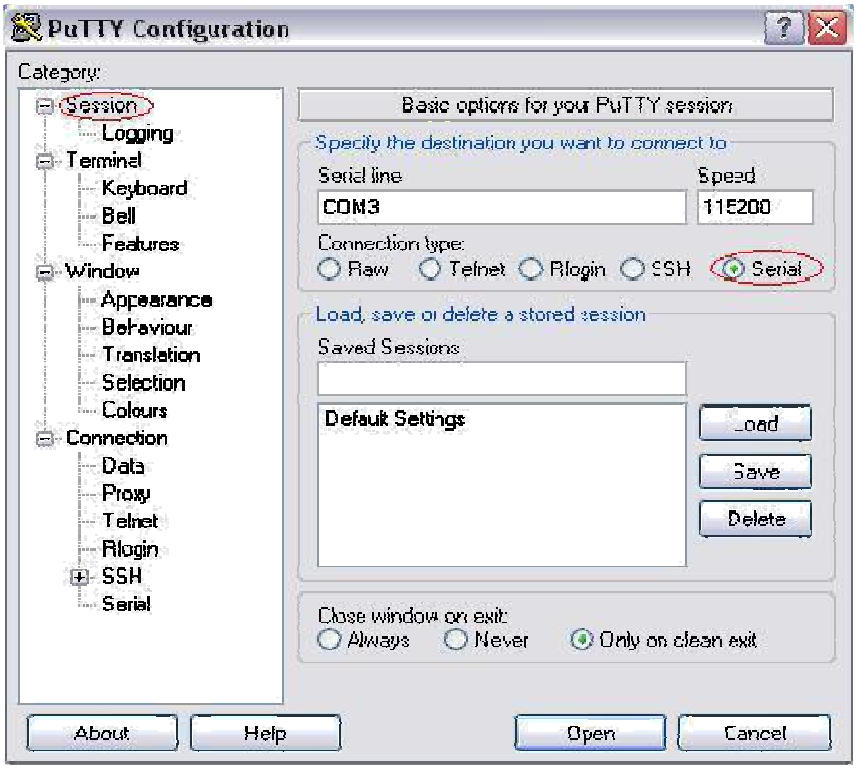


Рисунок 1.7 Интерфейс программы putty.exe

2. В категории *Translation* установите *UTF-8* и нажмите *Open* (рис. 1.8);

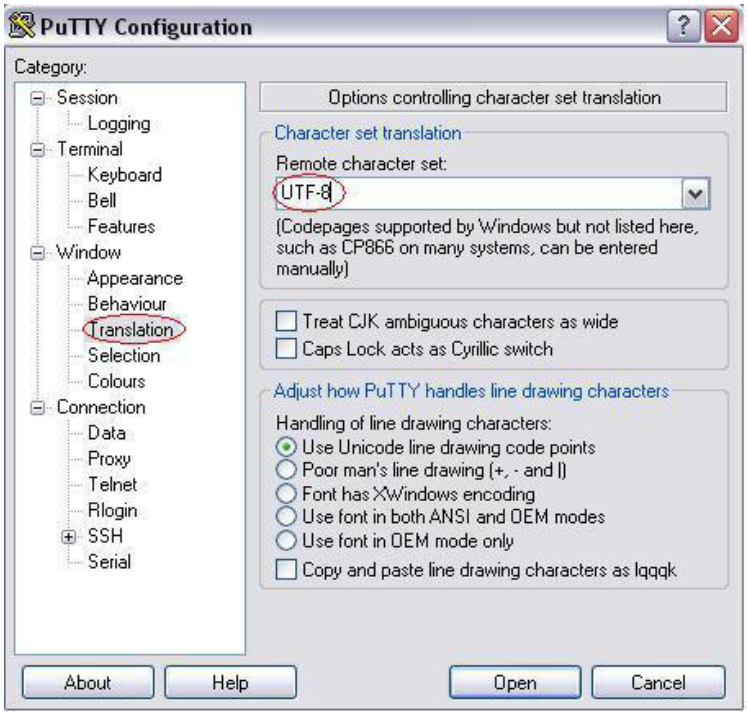


Рисунок 1.8 Интерфейс программы putty.exe

3. Нажмите кнопку *Open*. В открывшемся окне нажмите клавишу *Enter* (рис. 1.9).

Примечание: По умолчанию на коммутаторе UserName и PassWord не определены, поэтому два раза нажмите клавишу Enter.

После этого появится приглашение для ввода команд: DES-3200-10:admin#

Рисунок 1.9 Окно эмуляции терминала VT100

Внимание: при написании команд в CLI важно учитывать регистр. Для того чтобыознакомиться с правильностью написания команд, последовательностью выполнения операций можно обращаться к встроенной помощи по командам!

Примечание: не соединяйте порты коммутатора одним кабелем до особого указания.

Шаг 2. Проверьте доступность соединения между рабочими станциями ПК1 и ПК2.

В командной строке ПК1введите: ping 10.90.90.92

В командной строке ПК2 введите: ping 10.90.90.91

Шаг 3. Посмотрите загрузку портов

коммутатора:show utilization ports

Какая загрузка портов, используемых в схеме?

Порт 1% _____

Порт 2% _____

Порт 7% _____

Порт 8% _____

Шаг 4. Соберите схему и соедините кабелем Ethernet порты 1 и 7 коммутатора.

Шаг 5. Посмотрите загрузку портов

коммутатора:show utilization ports

Что вы наблюдаете? Возник широковещательный шторм? Почему?

_____

Какая теперь загрузка портов, используемых в схеме?

Порт 1% _____

Порт 2% _____

Порт 7% _____

Порт 8%_____

Шаг 6. Выполните на рабочей станции ПК1

команду:ping 10.90.90.92

Что вы наблюдаете? Объясните почему нет связи между рабочими станциями?

_____

Шаг 7. Удалите коммутационную петлю, отключив кабель от портов 1 и 7.

Шаг 8. Добавьте порты 2 и 8 в

новуюVLAN:config vlan default delete 2, 8

create vlan v2 tag 2

config vlan v2 add untagged 2,8

Шаг 9. Посмотрите загрузку портов

коммутатора:show utilization ports

Какая загрузка портов, используемых в схеме?

Порт 1% _____

Порт 2% _____

Порт 7% _____

Порт 8% _____

Шаг 10. Соедините кабелем Ethernet порты 1 и 7 коммутатора.

Шаг 11. Посмотрите загрузку портов

коммутатора:show utilization ports

Что вы наблюдаете? Почему нет широковещательного шторма на портах 2 и 6?

Шаг 12. Выполните на рабочей станции ПК1 команду:ping 10.90.90.92

Что вы наблюдаете? Объясните почему? _____

2.3 Лабораторная работа № 3 (2 часа)

Тема: «Линии связи»

2.3.1 Цель работы: изучить основные характеристики линий связи.

2.3.2 Задачи работы:

1. Ознакомиться с основными характеристиками каналов связи.

2.3.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. Персональный компьютер, включенный в сеть IP, Microsoft Windows.

2.3.4 Описание (ход) работы:

К основным характеристикам канала (линии) связи существенно влияющим на качество передачи сигнала можно отнести:

- полосу пропускания;
- затухание;
- помехоустойчивость;
- пропускную способность;
- достоверность передачи данных.

Полоса пропускания

Полоса пропускания (*bandwidth*) – диапазон частот, в пределах которого амплитудно-частотная характеристика (АЧХ) канала (линии) связи достаточно равномерна для того, чтобы обеспечить передачу сигнала без существенного искажения его формы.

Ширина полосы пропускания F определяется как разность верхней $f_в$ и нижней $f_н$ граничных частот участка АЧХ, на котором мощность сигнала уменьшается не более чем в 2 раза по сравнению с максимальным значением: $F=f_в - f_н$ (что приблизительно соответствует −3 дБ).

Измеряется полоса пропускания в герцах (Гц).

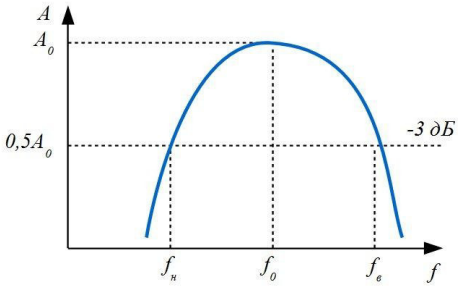


Рис. 1. Полоса пропускания канала связи

Ширина полосы пропускания существенным образом влияет на максимально возможную скорость передачи информации по каналу связи и зависит от типа среды передачи, наличия в каналах частотных фильтров.

Сигналы составлены из большого набора гармоник, однако приемник может получить лишь те гармоники, частоты которых находятся внутри полосы пропускания канала. Чем шире полоса пропускания канала, тем выше может быть скорость передачи данных и тем более высокочастотные гармоники сигнала могут передаваться. Если в полосу пропускания канала попадают гармоники, амплитуды которых вносят основной вклад в результирующий сигнал, форма сигнала претерпит незначительные изменения, и сигнал будет правильно распознан приемником.

В противном случае форма сигнала будет значительно искажаться, что приведет к снижению скорости передачи информации по каналу вследствие проблем с его распознаванием, которые вызовут ошибки связи и повторные передачи.

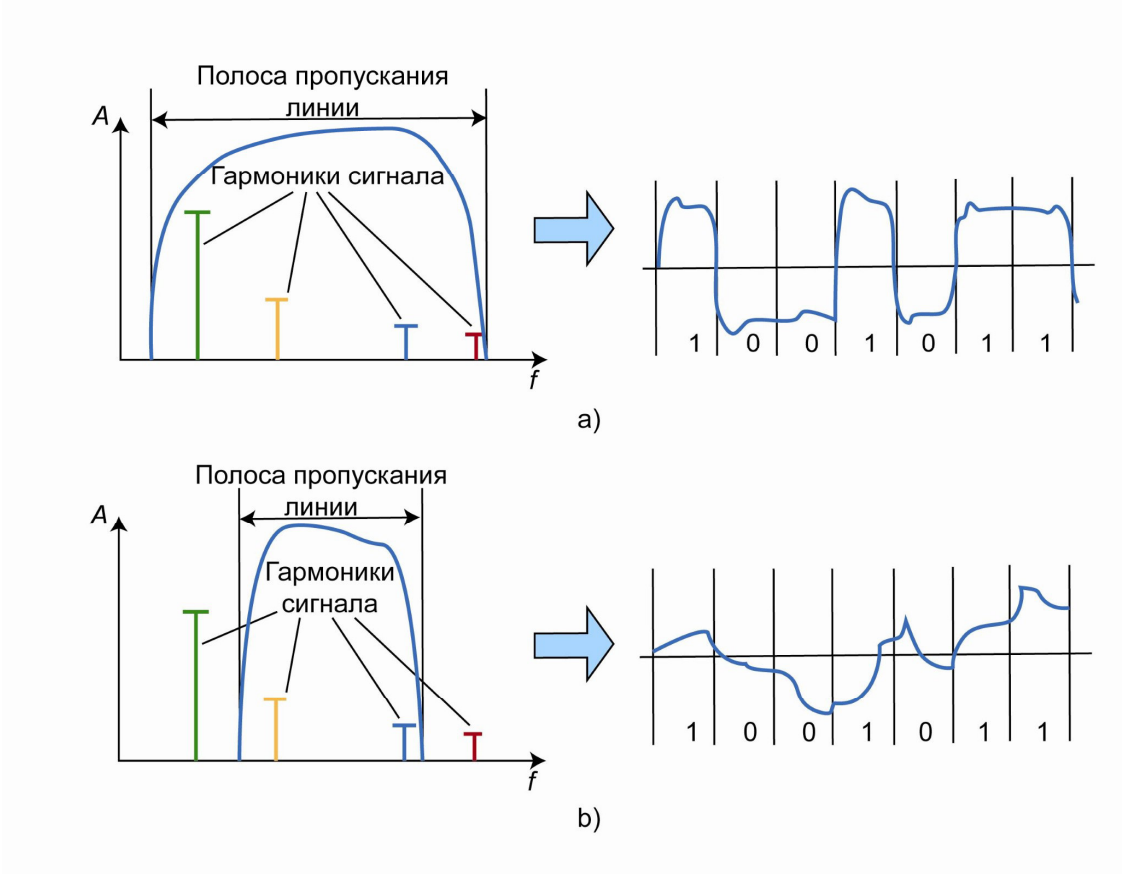


Рис. 2. Влияние полосы пропускания на сигнал

Затухание

При передаче сигнала по каналу связи, происходит его постепенное ослабление (*затухание*), что обусловлено физическими и техническими свойствами среды передачи и используемых сетевых устройств. Для корректного распознавания сигнала в точке приема это ослабление не должно превышать некоторой пороговой величины.

Затухание (*attenuation*) — это величина, показывающая, насколько уменьшается мощность (амплитуда) сигнала на выходе канала связи по отношению к мощности (амплитуде) сигнала на входе. Коэффициент затухания *d* измеряется в децибелах (дБ, dB) на единицу длины и вычисляется по следующей формуле:

$$d[\text{дБ}] = 10 \lg \frac{P_{вых}}{P_{вх}},$$

где $P_{вых}$ – мощность выходного сигнала; $P_{вх}$ – мощность входного сигнала.

Затухание характерно как для аналоговых, так и для цифровых сигналов. Оно увеличивается с ростом частоты сигнала: чем выше частота, тем сильнее сигнал подвержен затуханию. По этой причине приемникам высокоскоростного оборудования значительно сложнее распознать исходный сигнал.

Затухание сигнала влияет на расстояние, которое он может пройти между двумя точками без усиления или восстановления. Затухание является одним из важных параметров определенных для кабелей (витой пары, волоконно-оптического, коаксиального). Чем меньше затухание, тем более качественным является кабель. Поэтому при проектировании проводных каналов связи надо учитывать характеристики кабелей и использовать кабели с наименьшим значением затухания для достижения максимальной длины канала.

Помехоустойчивость

В реальном канале связи существуют помехи, обусловленные характеристиками среды передачи, каналообразующей аппаратуры, влиянием электромагнитных полей различных электронных устройств. В результате действия различных помех в канале связи появляются ошибки.

Одним из важнейших показателей канала связи является его **помехоустойчивость**, под которой понимают способность канала противостоять воздействию помех.
Помехоустойчивость основывается на возможности отличить сигнал от помехи с заданной достоверностью, поэтому при построении канала связи нужно учитывать возможные помехи и предельно использовать различие между ними и сигналом.

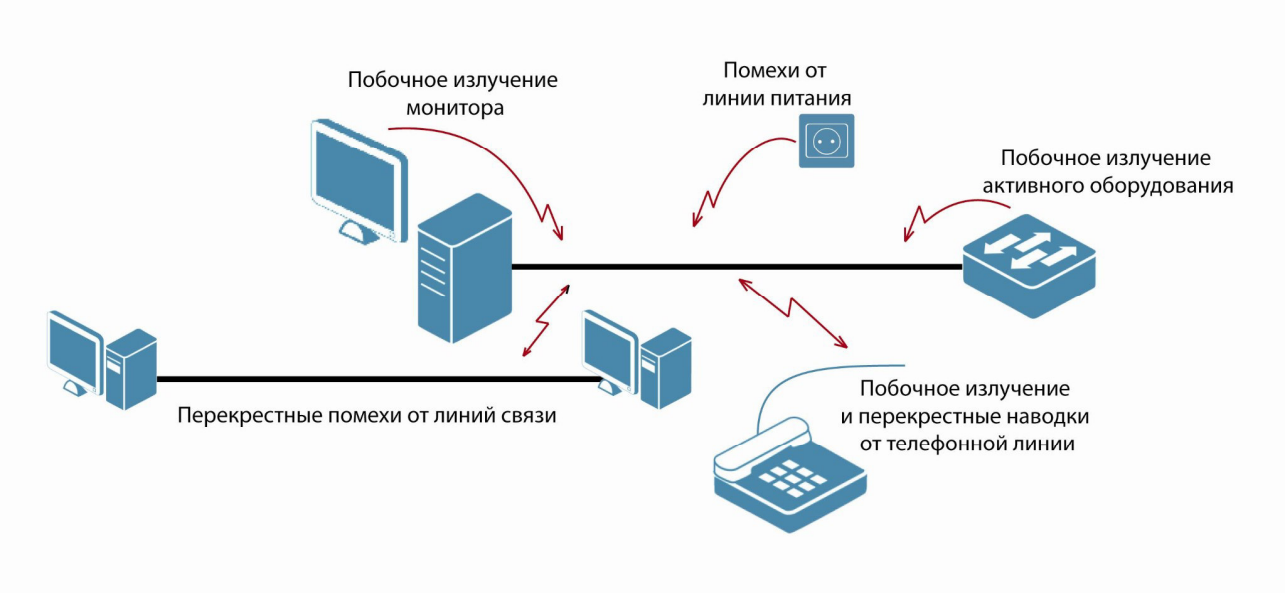


Рис. 4. Влияние помех на канал связи

В зависимости от источника возникновения и от характера их воздействия помехи делятся на внутренние, внешние и взаимные. *Внутренние помехи* или шумы возникают от источников находящихся в данном канале связи и появляются сразу же после включения оборудования связи. Они в основном определяются тепловыми, дробовыми, контактными и импульсными шумами и практически неустранимы.

Внешние помехи делятся на промышленные, радиопомехи, атмосферные и космические. Промышленные помехи (*электромагнитная интерференция*, Electro Magnetic Interference (EMI)) создаются в результате влияния на канал связи электромагнитных полей различных электрических устройств: ламп дневного света, бытовых приборов, компьютеров, радиосистем, линий электропередач, электрооборудования промышленных предприятий, медицинских установок, контактных сетей электрифицированного транспорта (трамвая, троллейбуса и т.п.), световой рекламы на газоразрядных лампах и т.п.

Радиопомехи (*радиочастотная интерференция*, Radio Frequency Interference (RFI)) возникают от излучения радиостанций различного назначения, спектр которых по какимлибо причинам накладывается на спектр полезных сигналов канала связи.

К атмосферным помехам относятся помехи, вызванные различными атмосферными явлениями: магнитными бурями, северными сияниями, грозовыми разрядами и т.д. К космическим помехам относятся электромагнитные помехи, создаваемые излучениями Солнца, видимых и невидимых звезд, туманностей в соответствующих диапазонах частот.

Взаимные (перекрестные, cross talk*)помехи* или наводки возникают при передаче информации по смежным каналам сигнал, переданный по одному каналу связи, создает нежелательный эффект в другом (возникает интерференция сигналов).

Наименее защищенными от влияния помех являются беспроводные каналы связи. На них действуют как внешние, так и перекрестные помехи. В беспроводных домашних сетях внешние помехи возникают от работающих микроволновых печей, компьютеров, сотовых телефонов и т.д. А перекрестные наводки связанны с помехами от другого беспроводного оборудования, работающего на той же частоте. Это особенно актуально в многоквартирных домах, где домашние сети в основном построены с использованием беспроводных технологий.

Среди кабельных каналов наиболее подвержены влиянию помех каналы на основе электрических кабелей. Для борьбы с помехами разработчики электрических кабелей используют: э*кранирование* (shielding) и *скручивание проводников*. Экранирование используется для защиты от электромагнитных и радиопомех. Экран представляет собой металлическую оплетку или фольгу, которая окружает каждый провод или группу проводов в кабеле. Он действует как барьер для взаимодействующих сигналов.

Электрические кабели сами являются источником электромагнитного излучения, которое может вызывать перекрестные помехи. В кабелях на основе витой пары эти помехи известны как *перекрестные наводки на ближнем конце* (Near End Cross Talk, NEXT) и *перекрестные наводки на дальнем конце* (Far End Cross Talk, FEXT) и связаны с взаимным влиянием электромагнитных полей сигналов, передаваемых по разным парам проводников. Для подавления этих электромагнитных полей используется скручивание проводников витой пары.

Наиболее защищенными от помех являются оптические каналы. На волоконнооптические кабели не воздействуют электромагнитные помехи (EMI), радиочастотные помехи (RFI), молнии и скачки высокого напряжения. Также волоконно-оптические кабели не создают никаких электромагнитных или радиочастотных помех.

Чтобы шумы заметно не снижали качества передачи их влияние необходимо ограничивать. Методы борьбы с шумами заключаются в обеспечении такого уровня сигнала в месте приема, который бы обеспечил требуемое качество принимаемого сигнала.

Одним из важных параметров канала связи, позволяющим оценить мешающее воздействие помех на сигнал является **отношение сигнал/шум** (*SNR, Signal-to-Noise Ratio*). Оно определяется как отношение мощности сигнала $P_с$ к мощности шума (помех) $P_ш$ и выражается в децибелах (дБ):

$$SRN\ [\text{дБ}] = 10 \lg\left(\frac{P_c}{P_ш}\right),$$

где P_c – мощность сигнала; $P_ш$ - мощность шума (помех).

При этом чем больше отношение сигнал/шум, тем меньше шум влияет на полезный сигнал при его передаче по каналу связи и ведет к хорошему распознаванию сигнала приемником.

Для повышения помехоустойчивости канала связи применяются следующие методы:

- увеличение отношения сигнал/шум;
- расширение спектра сигнала;
- увеличение избыточности информации;
- применение помехоустойчивых кодов;
- фильтрация полезного сигнала.

Пропускная способность

Пропускная способность (*throughput*) канала связи максимально возможная *информационная* скорость передачи данных количество данных, которое может быть передано по каналу связи за единицу времени. Измеряется пропускная способность в битах в секунду (бит/с или bps bits per second).

Максимальная пропускная способность зависит от полосы пропускания канала связи и отношения сигнал/шум и может быть рассчитана по формуле Клода Шеннона:

$$C = F \log_2\left(1 + \frac{P_c}{P_ш}\right),$$

где C – максимальная пропускная способность канала (бит/с); F – ширина полосы пропускания канала (Гц); P_c – мощность сигнала; $P_ш$ – мощность шума (помехи).

Как видно из формулы, пропускная способность канала может быть повышена за счет увеличения полосы пропускания F или увеличение отношения сигнал/шум. При этом первый способ более эффективен и менее трудоемок по сравнению со вторым, в связи с логарифмической зависимостьюC от P_c/ $P_ш$.

Реальная скорость передачиданных по каналу связи обычно *меньше* его *пропускной способности* и зависит от параметров каналообразующей аппаратуры, способов организации передачи данных, количества узлов, подключенных к каналу связи. Также на снижение скорости влияют накладные расходы, связанные с передачей по сети служебных сообщений, которые требуется для работы сетевых протоколов.

Следует понимать различие между информационной скоростью и символьной скоростью. *Информационная скорость* (information rate, bitrate) – это скорость передачи битов, измеряемая в бит/с и производных единицах. *Символьная скорость* (symbol rate) или *скорость модуляции* – это скорость изменения символов, измеряемая в бодах или символах в

секунду. Каждый символ представляет один или несколько битов информации в зависимости от выбранного способа их кодирования.

2.4 Лабораторная работа № 6 (2 часа)

Тема: «Протоколы и алгоритмы маршрутизации»

2.4.1 Цель работы: получить сведения о маршрутизации и научиться добавлять маршруты в таблицу маршрутизации.

2.4.2 Задачи работы:

1. Научиться работать с таблицей маршрутизацией.

2.4.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. Аппаратные: компьютер с установленной ОС Windows.
2. Программные: Приложения ВМ: VirtualBox; Виртуальные машины: VM-1.

2.4.4 Описание (ход) работы:

В сетях, основанных на протоколе IP, *концепция маршрутизации* является одной из важных. Она создает или разбивает сеть. Неправильная конфигурация маршрутизации способна вывести из строя сеть.

Маршрутизация – технология определения пути доставки (маршрута) пакетов. Основные принципы маршрутизации:

1. Каждая операционная система, поддерживающая стек **TCP/IP**, имеет маршрутизатор и таблицу маршрутизации.
2. Таблица маршрутизации используется только тогда, когда определяется, как доставлять пакеты.
3. Маршрутизация должна быть сконфигурирована корректно на обоих концах связи и на каждом участке между ними.

Для определения пути доставки пакета используется *таблица маршрутизации*. Пример таблицы маршрутизации можно получить командой **route** с параметром *print*.

Активные маршруты:

Сетевой адрес	Маска сети	Адрес шлюза	Интерфейс	Метрика
Network Destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.4.1	192.168.4.7	1
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.4.0	255.255.255.0	192.168.4.7	192.168.4.7	1
192.168.4.7	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.4.255	255.255.255.255	192.168.4.7	192.168.4.7	1
224.0.0.0	224.0.0.0	192.168.4.7	192.168.4.7	1
255.255.255.255	255.255.255.255	192.168.4.7	192.168.4.7	1
Основной шлюз:	192.168.1.1			

Рисунок 1. Пример таблицы маршрутизации

В общем случае для маршрутизации используется следующий *алгоритм*. Из пакета извлекается IP-адрес назначения пакета и производится попытка сопоставить его с адресом назначения (*Сетевой адрес*) каждого элемента таблицы маршрутизации пока не найдется наилучшее совпадение. Если совпадений не найдено, то пакет удаляется и отправителю пакета может отправиться сообщение об ошибке. Сравнение производится с тремя порциями информации: **Сетевой адрес (*Network Destination*), Маска сети (*Netmask*) и IP-адрес назначения пакета.**

В основном, производится побитная операция **AND** между **IP-адресом получателя** и **Маской сетии (*Netmask*)**: если полученное значение равно **Сетевому адресу (*Network Destination*)**, то считается, что совпадение найдено.

Пример 1. Необходимо проверить почту на сервере, чей адрес **192.168.4.100** (используется таблица маршрутизации приведенная ранее). Необходимо выполнить побитную операцию **AND** над **IP-адресом получателя пакетов** и **сетевыми масками (*Netmask*)** из таблицы маршрутизации. Эта операция производится над всем масками из таблицы маршрутизации. Но в рассматриваемом примере только 3-я строка наиболее походит.

	1-й октет								2-й октет								3-й октет								4-й октет							
биты	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ID-сети																													ID-узла		
IP-адрес	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1	0	0
десятичная запись	192								168								4								100							
Маска	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
десятичная запись	255								255								255								0							
Результат операции AND	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
десятичная запись	192								168								4								0							

Рисунок 2. Пример определения маршрута доставки пакетов

Как видно из приведенной таблицы, результат побитной операции **AND** совпадает с 3-й строкой таблицы маршрутизации (Рисунок 2). Следовательно, пакет отправится по указанному маршруту через интерфейс**192.168.4.7**.

Следует отметить, что указанный в примере IP-адрес после выполнения побитной операции **AND** над масками совпадет больше чем с одной строкой маршрутизации. Для избежания таких случаев используется*приоритет маршрутов*. Система ищет более точное совпадение адреса с маской (255.255.255.255 более точна, чем 255.255.255.0, которая в свою

очередь, более точна, чем 0.0.0.0). Маршрут с сетевым адресом 0.0.0.0 и маской 0.0.0.0 является *маршрутом по умолчанию*. Так как этот маршрут подходит к любому адресу назначения, он описывает маршрут, который используется, если не найден более подходящий. Обычно этот маршрут используется для пересылки пакетов провайдеру Интернет-услуг, при подключении к Интернету.

Для работы с таблицей маршрутизации используется стандартная утилита **ROUTE**, которая выводит на экран и изменяет записи в локальной таблице IP-маршрутизации.

Запущенная без параметров, команда **route** выводит справку.

Параметр	**Описание**
add	Добавление маршрута
change	Изменение существующего маршрута
delete	Удаление маршрута или маршрутов
print	Печать маршрута или маршрутов

*Таблица 1. Назначение параметров команды **route***

Пример 2. Добавление маршрута.

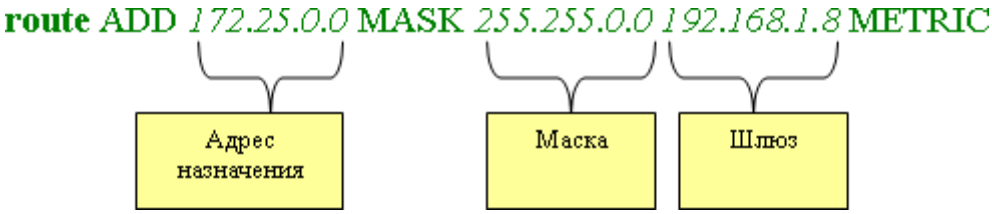


Рисунок 3. СТрока длядобавление маршрута

Задание 1. Создайте таблицу для облегчения определения маршрутов.

1. Откройте **табличный процессор** и сформируйте таблицу по следующему шаблону:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG
1		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
2	**IP-адрес**	192								168								252								56							
3	двоичная запись	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0
4	**Маска**	255								255								255								7							
5	двоичная запись	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1
6	**Операция AND**	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
7	**Адрес назначения**																																

Рисунок 4. Образец оформления таблицы

2. Введите в диапазон ячеек **Z3:AG3** формулы для перевода числа в десятичной системе счисления из ячейки **Z2** в двоичную форму (в соответствии с таблицей).

Таблица 2. Формулы для перевода в двоичную систему счисления

Имя Ячейки	**Формула**

AG3	=Z2-2*INT(Z2/2)
AF3	=INT(Z2/2)-2*INT(INT(Z2/2)/2)
AE3	=INT(INT(Z2/2)/2)-2*INT(INT(INT(Z2/2)/2)/2)
AD3	=INT(INT(INT(Z2/2)/2)/2)-2*INT(INT(INT(INT(Z2/2)/2)/2)/2)
AC3	=INT(INT(INT(INT(Z2/2)/2)/2)/2)-2*INT(INT(INT(INT(INT(Z2/2)/2)/2)/2)/2)
AB3	=INT(INT(INT(INT(INT(Z2/2)/2)/2)/2)/2)- 2*INT(INT(INT(INT(INT(INT(Z2/2)/2)/2)/2)/2)/2)
AA3	=INT(INT(INT(INT(INT(INT(Z2/2)/2)/2)/2)/2)/2)- 2*INT(INT(INT(INT(INT(INT(INT(Z2/2)/2)/2)/2)/2)/2)/2)
Z3	=INT(INT(INT(INT(INT(INT(INT(Z2/2)/2)/2)/2)/2)/2)/2)

3. Аналогично введите формулы для преобразования чисел из десятичной системы счисления в двоичную для ячеек **R2,J2,B2**.
4. Аналогично введите формулы для преобразования маски подсети в двоичную систему счисления.
5. Введите формулы для побитной операции **AND** над **IP-адресом** и **маской (*Netmask*)**:
   - введите в ячейку **AG6** формулу **=AND(AG3;AG5)**;
   - скопируйте введенную формулу в диапазон ячеек **B6:AF6**.
6. Введите в ячейку **Z7** формулу для преобразования 4-го октета маски в десятичную систему счисления - **=AG6*2^AL1+AF6*2^AF1+AE6*2^AE1+AD6*2^AD1+AC6*2^AC1+AB6*2^AB1+AA6*2^AA1+Z6*2^Z1**.
7. Аналогично введите формулы для ячеек **R7, J7, B8**.
8. Сохраните файл в своем каталоге с именем ***ROUTE***.

Задание 2. Создайте новый маршрут для вашего компьютера и проследите его.

1. Запустите виртуальную машину **VM-1** и загрузите ОС **Windows**.
2. Откройте **консоль (*Пуск/Программы/Стандартные/Командная строка*)**.
3. Определите IP-адрес вашего компьютера с помощью утилиты **ipconfig**.
4. Просмотрите таблицу маршрутизации на вашем компьютере:

- выведите справку по команде **route** (для этого необходимо ввести команду и нажать клавишу ***ENTER***);

```
Route
```

- выведите таблицу маршрутизации командой **route** с параметром *PRINT*:

```
route PRINT
```

- запомните маршрут по умолчанию (первая строка).

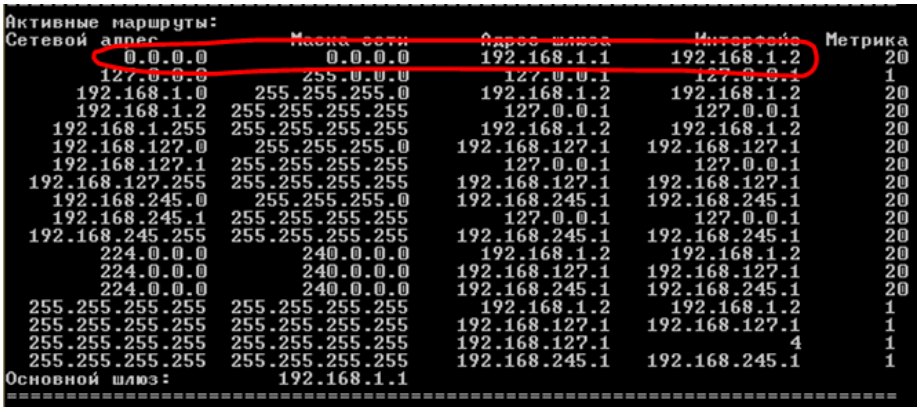


Рисунок 5.Пример вывода программы **ROUTE**

5. Проследите работу маршрутизатора с помощью утилиты *TRACERT*, отправив пакеты на узел **www.opennet.ru**. Введите: tracert www.opennet.ru

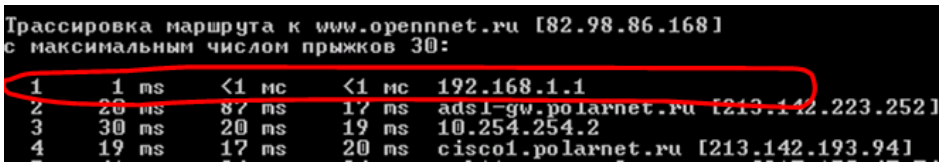


Рисунок 6. Пример вывода программы **TRACERT**

6. *Следует отметить, что пакеты на указанный сайт отправляются через один шлюз (**192.168.1.1**), который видно в первых строках вывода программ **ROUTE** и **TRACERT**.*
7. Добавьте в таблицу маршрутизации компьютера строку для пересылки пакетов в сеть **172.21.0.0** (маска **255.255.0.0**) через сетевой интерфейс компьютера. Введите:

```
route add 172.21.0.0 mask 255.255.0.0 192.168.1.4 METRIC 3
```

8. Проверьте работу внесенных вами изменений с помощью утилиты *TRACERT*.

Самостоятельные задания.

1. Сформируйте маски подсети таким образом, чтобы получались сети, в которых количество уникальных адресов составляют 256, 2048, 32768.

2. Определите маршруты для пакетов в соответствии с таблицей маршрутизации, приведенной в теоретической части лабораторной работы. Результат оформите в виде таблицы и сохраните в своей папке.

Таблица маршрутизации		
IP-адреса пакетов	Адрес шлюза	Интерфейс
10.1.1.1		
192.168.4.121		
127.13.13.210		
192.168.5.121		

2.5 Лабораторная работа № 5 (2 часа)

Тема: «Протоколы TCP/IP»

2.5.1 Цель работы: изучить эталонную модель протоколов ISO/OSI и стек протоколов TCP/IP. Изучить IP-адресацию и правила назначения IP-адресов. Познакомиться с протоколом IP, IP-адресацией, IP-маршрутизацией, протоколом TCP, функциями протокола TCP.

2.5.2 Задачи работы:

1. Изучить эталонную модель протоколов ISO/OSI и стек протоколов TCP/IP;
2. Изучить IP-адресацию и правила назначения IP-адресов.

2.5.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. Компьютерная ЛВС.

2.5.4 Описание (ход) работы:

Протокол – это набор правил, описывающих метод передачи информации по сети. Понятие протокола является исключительно важным для компьютерных сетей. Это связано с тем, что сеть может объединять компьютеры разных типов, работающие под управлением разных операционных систем. Чтобы эти компьютеры могли обмениваться друг с другом информацией, они должны «разговаривать на одном языке», то есть использовать одни и те же протоколы - правила передачи информации по сети.

Стек протоколов TCP/IP является протокольной основой Интернет. Ключевым моментом при этом является IP-адресация.

IP-адрес – это уникальный числовой адрес, однозначно идентифицирующий узел, группу узлов или сеть. IP-адрес имеет длину 4 байта и обычно записывается в виде четырех чисел (так называемых «октетов»), разделенных точками, каждое из которых может принимать значения в диапазоне от 0 до 255, например:

128.10.2.30 - традиционная десятичная форма представления адреса,

10000000 00001010 00000010 00011110 - двоичная форма представления этого же адреса.

Адрес состоит из двух логических частей - номера сети и номера узла в сети. Какая часть адреса относится к номеру сети, а какая к номеру узла, определяется значениями первых битов адреса:

- Если адрес начинается с 0, то сеть относят к классу A, и номер сети занимает один байт, остальные 3 байта интерпретируются как номер узла в сети. Сети класса A имеют номера в диапазоне от 1 до 126. (Номер 0 не используется, а номер 127

зарезервирован для специальных целей). В сетях класса A количество узлов должно быть больше 2^{16}, но не превышать 2^{24}.

- Если первые два бита адреса равны 10, то сеть относится к классу B и является сетью средних размеров с числом узлов 2^8 - 2^{16}. В сетях класса B под адрес сети и под адрес узла отводится по 16 битов, то есть по 2 байта.
- Если адрес начинается с последовательности 110, то это сеть класса C с числом узлов не больше 2^8. Под адрес сети отводится 24 бита, а под адрес узла - 8 битов.
- Если адрес начинается с последовательности 1110, то он является адресом класса D и обозначает особый, групповой адрес - multicast. Если в пакете в качестве адреса назначения указан адрес класса D, то такой пакет должны получить все узлы, которым присвоен данный адрес.
- Если адрес начинается с последовательности 11110, то это адрес класса E, он зарезервирован для будущих применений.

В таблице приведены диапазоны номеров сетей, соответствующих каждому классу сетей.

Класс	Наименьший адрес	Наибольший адрес
A	1.0.0.0	126.0.0.0
B	128.0.0.0	191.255.0.0
C	192.0.1.0	223.255.255.0
D	224.0.0.0	239.255.255.255
E	240.0.0.0	247.255.255.255

В протоколе IP существует несколько соглашений об особой интерпретации IP-адресов:

- если IP-адрес состоит только из двоичных нулей, то он обозначает адрес того узла, который сгенерировал этот пакет;

0 0 0 0 ..... 0 0 0 0

- если в поле номера сети стоят 0, то по умолчанию считается, что этот узел принадлежит той же самой сети, что и узел, который отправил пакет;

0 0 0 0 .....0 Номер узла

- если все двоичные разряды IP-адреса равны 1, то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета. Такая рассылка называется ограниченным широковещательным сообщением (limited broadcast);

1 1 1 1 .....1 1

- если в поле адреса назначения стоят сплошные 1, то пакет, имеющий такой адрес рассылается всем узлам сети с заданным номером. Такая рассылка называется широковещательным сообщением (broadcast);

Номер сети 1111.....11

- адрес 127.0.0.1 зарезервирован для организации обратной связи при тестировании работы программного обеспечения узла без реальной отправки пакета по сети. Этот адрес имеет название loopback.

Уже упоминавшаяся форма группового IP-адреса - multicast - означает, что данный пакет должен быть доставлен сразу нескольким узлам, которые образуют группу с номером, указанным в поле адреса. Узлы сами идентифицируют себя, то есть определяют, к какой из групп они относятся. Один и тот же узел может входить в несколько групп. Такие сообщения в отличие от широковещательных называются мультивещательными. Групповой адрес не делится на поля номера сети и узла и обрабатывается маршрутизатором особым образом.

1. Ознакомиться с теоретическими сведениями по теме. Особенно внимательно изучить материал, относящийся к IP-адресации.
2. На основе примера, разобранного для сетей класса А, заполнить третью колонку таблицы 1.
3. Выполнить аналогичные расчеты и заполнить четвертую и пятую колонки таблицы 1.

Для выполнения задания 2 необходимо выполнить следующие действия:

1. Перевести каждое число IP-адреса в двоичную форму. Для перевода можно воспользоваться программой «Калькулятор», установив «Вид/Инженерный».
2. По первым битам IP-адреса определить класс сети.
3. В соответствии с классом определить маску сети по умолчанию.
4. Выписать только те биты IP-адреса, которые соответствуют единичным битам в маске сети. Представить эти биты в точечной нотации. Это будет номер сети.
5. Выписать те биты IP-адреса, которые соответствуют нулевым битам в маске сети. Представить их в точечной нотации. Это будет номер хоста.
6. В двоичном представлении IP-адреса биты, соответствующие номеру хоста, заменить единицами. Представить получившийся адрес в точечной нотации. Это будет широковещательный адрес.

Задание

1. Ознакомьтесь с теоретическими сведениями по теме «Протоколы. IP-адресация».
2. Заполните таблицу 1 «Характеристики сетей различных классов».

Таблица 1

Номер по порядку	Характеристика сети	Класс сети		
		A	B	C
1	2	3	4	5
1.	Формат первого байта IP-адреса			
2.	Число байтов для номера сети			
3.	Число байтов для номера хоста			
4.	Минимальный номер сети в точечной нотации			
5.	Максимальный номер сети в точечной нотации			
6.	Число различных сетей			
7.	Минимальный номер хоста в точечной нотации			
8.	Максимальный номер хоста в точечной нотации			
9.	Число различных хостов			
10.	Маска сети по умолчанию			

3. Для IP-адреса, указанного в индивидуальном задании, считая, что маска сети задана по умолчанию, определите:
    3.1. Класс сети;
    3.2. Число сетей;
    3.3. Маску сети по умолчанию;
    3.4. Номер сети;
    3.5. Номер хоста;
    3.6. Минимальный номер сети;
    3.7. Максимальный номер сети;
    3.8. Широковещательный адрес.
4. Используя маску, указанную в индивидуальном задании, определите
    4.1. Маску сети (в десятичной нотации);
    4.2. Номер сети (в десятичной нотации);
    4.3. Номер хоста (в десятичной нотации);
    4.4. Минимальный номер хоста;
    4.5. Максимальный номер хоста;
    4.6. Широковещательный адрес;
    4.7. Число хостов.

Пример выполнения задания 2.

Пусть IP-адрес 64.10.20.30

Переводим числа в двоичный формат:

$64_{10}=01000000_2$

$10_{10}=00001010_2$

$20_{10}=00010100_2$

$30_{10}=00011110_2$

Записываем двоичную форму представления IP-адреса:

01000000.00001010.00010100.00011110

Первые биты адреса – 01, значит, это сеть класса А.

Маска сети по умолчанию: 255.0.0.0

Записываем в двоичной форме маску сети и IP-адрес:

Маска: 11111111. 00000000.00000000.00000000

IP-адрес: 01000000. 00001010.00010100.00011110

Эти биты соответствуют номеру сети

А эти биты соответствуют номеру хоста

Значит, номер сети - 01000000_2 или 64_{10}

номер хоста - $00001010.00010100.00011110_2$ или $10.20.30_{10}$

Заменяем в IP-адресе номер хоста единицами, получим широковещательный адрес $01000000.111111.111111.111111_2$ или 64.255.255.255

Следовательно:

IP-адрес	64.10.20.30
Класс сети	A
Маска сети	255.0.0.0
Номер сети	64.0.0.0
Номер хоста	0.10.20.30
Широковещательный адрес	64.255.255.255
Число сетей	2^7-2 =

При выполнении задания 3 необходимо вначале определить маску сети. Маска содержит столько единичных битов, сколько указано в числе после дробной черты. Остальные вычисления выполняются подобно заданию 2.

Контрольные вопросы

- Что такое протокол?
- Назовите уровни модели протоколов модели ISO/OSI и назначение протоколов каждого уровня.
- Назовите уровни стека протоколов TCP/IP и назначение протоколов каждого уровня.
- Приведите примеры протоколов, входящих в стек TCP/IP.

- Что такое аппаратный адрес?
- Что такое IP-адрес?
- Каковы правила назначения IP-адресов?
- Как проанализировать IP-адрес?

Варианты индивидуальных заданий

Таблица 2

Номер варианта	IP-адрес к заданию 3	IP-адрес к заданию 4
1.	192.168.72.33	192.168.72.33/20
2.	190.172.55.40	190.172.55.40/25
3.	123.232.14.72	123.232.14.72/18
4.	196.232.66.54	196.232.66.54/25
5.	193.123.55.67	193.123.55.67/26
6.	191.172.55.42	191.172.55.42/27
7.	178.66.57.18	178.66.57.18/20
8.	10.0.0.20	10.0.0.20/12
9.	67.192.44.89	67.192.44.89/12
10.	128.34.67.11	128.34.67.11/18
11.	193.34.126.44	193.34.126.44/26
12.	156.32.11.93	156.32.11.93/23
13.	167.168.169.170	167.168.169.17/20
14.	145.44.11.77	145.44.11.77/22
15.	132.45.171.99	132.45.171.99/25
16.	198.164.55.55	198.164.55.55/26
17.	192.77.121.144	192.77.121.144/25
18.	12.13.14.15	12.13.14.15/18
19.	44.57.62.39	44.57.62.39/18
20.	152.15.66.5	152.15.66.5/26
21.	132.45.171.99	132.45.171.99/27
22.	198.164.155. 5	198.164.155.5/26
23.	192.77.11.44	192.77.11.44/29
24.	12.130.140.150	12.130.140.150/17
25.	44.57.162.31	44.57.162.31/18
26.	152.154.66.65	152.154.66.65/20
27.	152.15.66.17	152.15.66.17/22
28.	132.45.171.88	132.45.171.88/21

Заключение: Выполнив эту практическую работу, Вы узнаете, каков формат IP-адреса, что такое маска сети, научитесь выделять составные части IP-адреса и определять по нему класс сети.

Протокол IP

Основу транспортных средств стека протоколов TCP/IP составляет протокол межсетевого взаимодействия *(Internet Protocol, IP).* Он обеспечивает передачу дейтаграмм от отправителя к получателям через объединенную систему компьютерных сетей.

Название данного протокола - *Internet Protocol* - отражает его суть: он должен передавать пакеты между сетями. В каждой очередной сети, лежащей на пути перемещения пакета, протокол IP вызывает средства транспортировки, принятые в этой сети, чтобы с их помощью передать этот пакет на маршрутизатор, ведущий к следующей сети, или непосредственно на узел-получатель.

Протокол IP относится к протоколам без установления соединений. Перед IP не ставится задача надежной доставки сообщений от отправителя к получателю. Протокол IP обрабатывает каждый IP-пакет как независимую единицу, не имеющую связи ни с какими другими IP-пакетами. В протоколе IP нет механизмов, обычно применяемых для увеличения достоверности конечных данных: отсутствует квитирование - обмен подтверждениями между отправителем и получателем, нет процедуры упорядочивания, повторных передач или других подобных функций. Если во время продвижения пакета произошла какая-либо ошибка, то протокол IP по своей инициативе ничего не предпринимает для исправления этой ошибки. Например, если на промежуточном маршрутизаторе пакет был отброшен по причине истечения времени жизни или из-за ошибки в контрольной сумме, то модуль IP не пытается заново послать испорченный или потерянный пакет. Все вопросы обеспечения надежности доставки данных по составной сети в стеке TCP/IP решает протокол TCP, работающий непосредственно над протоколом IP. Именно TCP организует повторную передачу пакетов, когда в этом возникает необходимость.

Важной особенностью протокола IP, отличающей его от других сетевых протоколов (например, от сетевого протокола IPX), является его способность выполнять динамическую фрагментацию пакетов при передаче их между сетями. Свойство фрагментации во многом способствовало тому, что протокол IP смог занять доминирующие позиции в сложных составных сетях.

Имеется прямая связь между функциональной сложностью протокола и сложностью заголовка пакетов, которые этот протокол использует. Это объясняется тем, что основные служебные данные, на основании которых протокол выполняет то или иное действие, переносятся между двумя модулями, реализующими этот протокол на разных машинах,

именно в полях заголовков пакетов. Поэтому очень полезно изучить назначение каждого поля заголовка IP-пакета, и это изучение дает не только формальные знания о структуре пакета, но и объясняет все основные режимы работы протокола по обработке и передаче IP-дейтаграмм.

IP-адресация

Компьютер в сети может иметь адреса трех уровней: физический (MAC-адрес), сетевой (IP-адрес) и доменный адрес (DNS-имя).

- Локальный адрес узла, определяемый технологией, с помощью которой построена отдельная сеть, в которую входит данный узел. Для узлов, входящих в локальные сети - это MAC - адрес сетевого адаптера или порта маршрутизатора, например, 11-A0-17-3D-BC-01. Эти адреса назначаются производителями оборудования и являются уникальными адресами, так как управляются централизовано. Для всех существующих технологий локальных сетей MAC-адрес имеет формат 6 байтов: старшие 3 байта - идентификатор фирмы производителя, а младшие 3 байта назначаются уникальным образом самим производителем. Для узлов, входящих в глобальные сети, такие как X.25 или frame relay, локальный адрес назначается администратором глобальной сети.

- IP-адрес, используется на сетевом уровне. Он назначается администратором во время конфигурирования компьютеров и маршрутизаторов. IP-адрес состоит из двух частей: номера сети и номера узла, например, 109.26.17.100.

Номер сети может быть выбран администратором произвольно, либо назначен по рекомендации специального подразделения Internet (Network Information Center, NIC), если сеть должна работать как составная часть Internet. Обычно провайдеры услуг Internet получают диапазоны адресов у подразделений NIC, а затем распределяют их между своими абонентами.

Номер узла в протоколе IP назначается независимо от локального адреса сети. Деление IP-адреса на поле номера сети и номера узла - гибкое, и граница между этими полями может устанавливаться весьма произвольно. Узел может входить в несколько IP-сетей. В этом случае узел должен иметь несколько IP-адресов, по числу сетевых связей. Таким образом, IP-адрес характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение.

– Символьный идентификатор-имя, например, SERV1.IBM.COM. Этот адрес назначается администратором и состоит из нескольких частей, например, имени машины, имени организации, имени домена. Такой адрес, называемый также доменным именем, используется на прикладном уровне, например, в протоколах FTP или telnet.

Символьные доменные имена. Символьные имена в IP-сетях называются доменными и строятся по иерархическому признаку. Составляющие полного символьного имени в IP-сетях

разделяются точкой и перечисляются в следующем порядке: сначала простое имя конечного узла, затем имя группы узлов (например, имя организации), затем имя более крупной группы (поддомена) и так до имени домена самого высокого уровня (например, домена объединяющего организации по географическому принципу:RU - Россия, UK - Великобритания, SU - США), Примеров доменного имени может служить имя base2.sales.zil.ru. Между доменным именем и IP-адресом узла нет никакого алгоритмического соответствия, поэтому необходимо использовать какие-то дополнительные таблицы или службы, чтобы узел сети однозначно определялся как по доменному имени, так и по IP-адресу. В сетях TCP/IP используется специальная распределенная служба Domain Name System (DNS), которая устанавливает это соответствие на основании создаваемых администраторами сети таблиц соответствия. Поэтому доменные имена называют также DNS-именами,

Классы IP-адресов. Для организации всемирной сети нужна хорошая система адресации, которая будет использоваться для направления информации всем адресатам. Союз Internet установил для адресации всех узлов Internet единый стандарт, называемый адресацией IP. Любой IP-адрес состоит из четырех чисел в интервале от 1 до 254, разделенных точками. Ниже приведен пример IP-адреса: 10.18.49.102. В схемах IP-адресации также могут использоваться числа 0 и 255, но они зарезервированы для специальных целей. Число 255 используется для направления дейтаграммы всем компьютерам сети IP. Число 0 используется для более точного указания адреса. Предположим, что в приведенном выше примере адрес служит для обозначения узла 102 в сети 10.18.49.102. В таком случае адрес 10.18.49.0 будет обозначать только сеть, а 0.0.0.102 будет обозначать один узел.

IP-адрес можно использовать для построения как сетей с несколькими узлами, так и сетей, содержащих миллионы узлов. IP-адрес имеет длину 4 байта и обычно записывается в виде четырех чисел, представляющих значения каждого байта в десятичной форме и разделенных точками, например, 128.10.2.30 - традиционная десятичная форма представления адреса, а 10000000 00001010 00000010 00011110 - двоичная форма представления этого же адреса.

Адрес состоит из двух логических частей - номера сети и номера узла в сети. Какая часть адреса относится к номеру сети, а какая - к номеру узла, определяется значениями первых бит адреса. Значения этих бит являются также признаками того, к какому *классу* относится тот или иной IP-адрес.

На рисунке 1 показана структура IP-адреса разных классов.

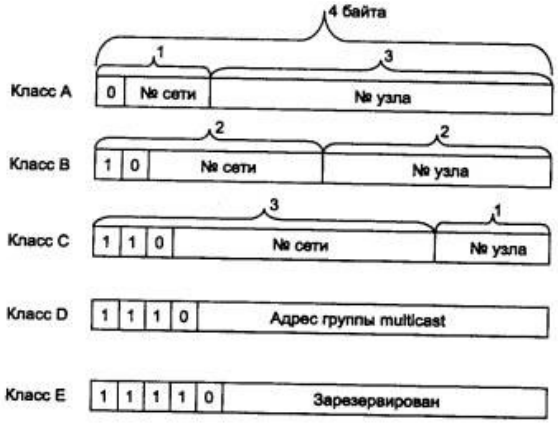


Рисунок 1 - Структура IP-адреса

Если адрес начинается с 0, то сеть относят к *классу A* и номер сети занимает один байт, остальные 3 байта интерпретируются как номер узла в сети. Сети класса A имеют номера в диапазоне от 1 до 126. (Номер 0 не используется, а номер 127 зарезервирован для специальных целей). Сетей класса A немного, зато количество узлов в них может достигать 2^{24}, то есть 16 777 216 узлов.

Если первые два бита адреса равны 10, то сеть относится к *классу B*. В сетях класса Впод номер сети и под номер узла отводится по 16 бит, то есть по 2 байта. Таким образом, сеть класса B является сетью средних размеров с максимальным числом узлов 2^{16}, что составляет 65 536 узлов.

Если адрес начинается с последовательности 110, то это сеть *класса C*. В этом случае под номер сети отводится 24 бита, а под номер узла - 8 бит. Сети этого класса наиболее распространены, число узлов в них ограничено 28, то есть 256 узлами.

Если адрес начинается с последовательности 1110, то он является адресом *класса D и* обозначает особый, групповой адрес - multicast. Если в пакете в качестве адреса назначения указан адрес класса D, то такой пакет должны получить все узлы, которым присвоен данный адрес.

Если адрес начинается с последовательности 11110, то это значит, что данный адрес относится к *классу E,* Адреса этого класса зарезервированы для будущих применений.

В таблице 1 приведены диапазоны номеров сетей и максимальное число узлов, соответствующих каждому классу сетей.

Таблица 1- Характеристики адресов разного класса

Класс	Первые биты	Наименьший номер сети	Наибольший номер сети	Максимальное число узлов в сети
A	0	1.0.0.0	126.0.0.0	2^{24}

B	10	128.0.0.0	191.255.0.0	2^{16}
C	110	192.0.1.0	223.255.255.0	2^{8}
D	1110	224.0.0.0	239.255.255.255	Multicast
E	11110	240.0.0.0	247.255.255.255	Зарезервирован

Большие сети получают адреса класса A, средние - класса B, а маленькие класса C.

Особые IP-адреса. В протоколе IP существует несколько соглашений об особой интерпретации IP-адресов.

- Если весь IP-адрес состоит только из двоичных нулей, то он обозначает адрес того узла, который сгенерировал этот пакет; этот режим используется только в некоторых сообщениях ICMP.

- Если в поле номера сети стоят только нули, то по умолчанию считается, что узел назначения принадлежит той же самой сети, что и узел, который отправил пакет.

- Если все двоичные разряды IP-адреса равны 1, то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета. Такая рассылка называется *ограниченным широковещательным.сообщением (limited broadcast)*.

- Если в поле номера узла назначения стоят только единицы, то пакет, имеющий такой адрес, рассылается всем узлам сети с заданным номером сети. Например, пакет с адресом 192.190.21.255 доставляется всем узлам сети 192.190.21.0. Такая рассылка называется *широковещательным сообщением (broadcast)*.

При адресации необходимо учитывать те ограничения, которые вносятся особым назначением некоторых IP-адресов. Так, ни номер сети, ни номер узла не может состоять только из одних двоичных единиц или только из одних двоичных нулей. Отсюда следует, что максимальное количество узлов, приведенное в таблице для сетей каждого класса, на практике должно быть уменьшено на 2. Например, в сетях класса C под номер узла отводится 8 бит, которые позволяют задавать 256 номеров: от 0 до 255. Однако на практике максимальное число узлов в сети класса C не может превышать 254, так как адреса 0 и 255 имеют специальное назначение. Из этих же соображений следует, что конечный узел не может иметь адрес типа 98.255.255.255, поскольку номер узла в этом адресе класса A состоит из одних двоичных единиц.

Особый смысл имеет IP-адрес, первый октет которого равен 127. Он используется для тестирования программ и взаимодействия процессов в пределах одной машины. Когда программа посылает данные по IP-адресу 127.0.0.1, то образуется как бы «петля». Данные не передаются по сети, а возвращаются модулям верхнего уровня, как только что принятые. Поэтому в IP-сети запрещается присваивать машинам IP-адреса, начинающиеся со 127. Этот

адрес имеет название *loopback*. Можно отнести адрес 127.0.0.0 ко внутренней сети модуля маршрутизации узла, а адрес 127.0.0.1 - к адресу этого модуля на внутренней сети. На самом деле любой адрес сети 127.0.0.0 служит для обозначения своего модуля маршрутизации, а не только 127.0.0.1, например 127.0.0.3.

В протоколе IP нет понятия широковещательности в том смысле, в котором оно используется в протоколах канального уровня локальных сетей, когда данные должны быть доставлены абсолютно всем узлам. Как ограниченный широковещательный IP-адрес, так и широковещательный IP-адрес имеют пределы распространения в интерсети - они ограничены либо сетью, к которой принадлежит узел-источник пакета, либо сетью, номер которой указан в адресе назначения. Поэтому деление сети с помощью маршрутизаторов на части локализует широковещательный шторм пределами одной из составляющих общую сеть частей просто потому, что нет способа адресовать пакет одновременно всем узлам всех сетей составной сети.

Уже упоминавшаяся форма группового IP-адреса - *multicast* - означает, что данный пакет должен быть доставлен сразу нескольким узлам, которые образуют группу с номером, указанным в поле адреса. Узлы сами идентифицируют себя, то есть определяют, к какой из групп они относятся. Один и тот же узел может входить в несколько групп. Члены какой-либо группы multicast не обязательно должны принадлежать одной сети. В общем случае они могут распределяться по совершенно различным сетям, находящимся друг от друга на произвольном количестве хопов. Групповой адрес не делится на поля номера сети и узла и обрабатывается маршрутизатором особым образом.

Основное назначение multicast-адресов - распространение информации по схеме «один-ко-многим». Хост, который хочет передавать одну и ту же информацию многим абонентам, с помощью специального протокола IGMP (Internet Group Management Protocol) сообщает о создании в сети новой мультивещательной группы с определенным адресом. Машрутизаторы, поддерживающие мультивещательность, распространяют информацию о создании новой группы в сетях, подключенных к портам этого маршрутизатора. Хосты, которые хотят присоединиться к вновь создаваемой мультивещательной группе, сообщают об этом своим локальным маршрутизаторам и те передают эту информацию хосту, инициатору создания новой группы.

Чтобы маршрутизаторы могли автоматически распространять пакеты с адресом multicast по составной сети, необходимо использовать в конечных маршрутизаторах модифицированные протоколы обмена маршрутной информацией, такие как, например, MOSPF (Multicast OSPF, аналог OSPF).

Групповая адресация предназначена для экономичного распространения в Internet или большой корпоративной сети аудио- или видеопрограмм, предназначенных сразу большой

аудитории слушателей или зрителей. Если такие средства найдут широкое применение (сейчас они представляют в основном небольшие экспериментальные островки в общем Internet), то Internet сможет создать серьезную конкуренцию радио и телевидению.

Использование масок в IP-адресации. Традиционная схема деления IP-адреса на номер сети и номер узла основана на понятии класса, который определяется значениями нескольких первых бит адреса. Именно потому, что первый байт адреса 185.23.44.206 попадает в диапазон 128-191, мы можем сказать, что этот адрес относится к классу B, а значит, номером сети являются первые два байта, дополненные двумя нулевыми байтами - 185.23.0.0, а номером узла - 0.0.44.206.

А что если использовать какой-либо другой признак, с помощью которого можно было бы более гибко устанавливать границу между номером сети и номером узла? В качестве такого признака сейчас получили широкое распространение маски. *Маска* - это число, которое используется в паре с IP-адресом; двоичная запись маски содержит единицы в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети. Поскольку номер сети является цельной частью адреса, единицы в маске также должны представлять непрерывную последовательность.

Для стандартных классов сетей маски имеют следующие значения:

- класс A - 11111111. 00000000. 00000000. 00000000 (255.0.0.0);
- класс B - 11111111. 11111111. 00000000. 00000000 (255.255.0.0);
- класс C-11111111.11111111.11111111.00000000 (255.255.255.0).

Снабжая каждый IP-адрес маской, можно отказаться от понятий классов адресов и сделать более гибкой систему адресации. Например, если рассмотренный выше адрес 185.23.44.206 ассоциировать с маской 255.255.0.0, то номером сети будет 185.23.44.0, а не 185.23.0.0, как это определено системой классов.

В масках количество единиц в последовательности, определяющей границу номера сети, не обязательно должно быть кратным 8, чтобы повторять деление адреса на байты. Пусть, например, для IP-адреса 129.64.134.5 указана маска 255.255.128.0, то есть в двоичном виде:

IP-адрес 129.64.134.5 - 10000001. 01000000.10000110. 00000101

Маска 255.255.128.0 - 11111111.11111111.10000000. 00000000

Если игнорировать маску, то в соответствии с системой классов адрес 129.64.134.5 относится к классу B, а значит, номером сети являются первые 2 байта - 129.64.0.0, а номером узла - 0.0.134.5.

Если же использовать для определения границы номера сети маску, то 17 последовательных единиц в маске, «наложенные» на IP-адрес, определяют в качестве номера сети в двоичном выражении число:

10000001. 01000000. 10000000. 00000000 или в десятичной форме записи - номер сети 129.64.128.0, а номер узла 0.0.6.5.

Механизм масок широко распространен в IP-маршрутизации, причем маски могут использоваться для самых разных целей. С их помощью администратор может структурировать свою сеть, не требуя от поставщика услуг дополнительных номеров сетей. На основе этого же механизма поставщики услуг могут объединять адресные пространства нескольких сетей путем введения так называемых «префиксов» с целью уменьшения объема таблиц маршрутизации и повышения за счет этого производительности маршрутизаторов.

Отображение IP-адресов на локальные адреса. Протокол ARP

Для отображения IP-адресов в Ethernet адреса используется протокол ARP (Address Resolution Protocol - адресный протокол). Отображение выполняется только для отправляемых IP-пакетов, так как только в момент отправки создаются заголовки IP и Ethernet.

ARP-таблица для преобразования адресов.Преобразование адресов выполняется путем поиска в таблице. Эта таблица, называемая ARP-таблицей, хранится в памяти и содержит строки для каждого узла сети. В двух столбцах содержатся IP- и Ethernet-адреса. Если требуется преобразовать IP-адрес в Ethernet-адрес, то ищется запись с соответствующим IP-адресом. Ниже приведен пример упрощенной ARP-таблицы.

Принято все байты 4-байтного IP-адреса записывать десятичными числами, разделенными точками. При записи 6-байтного Ethernet-адреса каждый байт указывается в 16-ричной системе и отделяется двоеточием.

Таблица 2 - Пример ARP-таблицы

IP-адрес	Ethernet-адрес
223.1.2.1	08:00:39:00:2F:C3
223.1.2.3	08:00:5A:21:A7:22
223.1.2.4	08:00:10:99:AC:54

ARP-таблица необходима потому, что IP-адреса и Ethernet-адреса выбираются независимо, и нет какого-либо алгоритма для преобразования одного в другой. IP-адрес выбирает менеджер сети с учетом положения машины в сети internet. Если машину перемещают в другую часть сети internet, то ее IP-адрес должен быть изменен. Ethernet-адрес выбирает производитель сетевого интерфейсного оборудования из выделенного для него по лицензии адресного пространства. Когда у машины заменяется плата сетевого адаптера, то меняется и ее Ethernet-адрес.

Порядок преобразования адресов. В ходе обычной работы сетевая программа, такая как TELNET, отправляет прикладное сообщение, пользуясь транспортными услугами TCP. Модуль TCP посылает соответствующее транспортное сообщение через модуль IP. В результате составляется IP-пакет, который должен быть передан драйверу Ethernet. IP-адрес места назначения известен прикладной программе, модулю TCP и модулю IP. Необходимо на его основе найти Ethernet-адрес места назначения. Для определения искомого Ethernet-адреса используется ARP-таблица.

Запросы и ответы протокола ARP. Как же заполняется ARP-таблица? Она заполняется автоматически модулем ARP, по мере необходимости. Когда с помощью существующей ARP-таблицы не удается преобразовать IP-адрес, то происходит следующее:

1) По сети передается широковещательный ARP-запрос.
2) Исходящий IP-пакет ставится в очередь.

Каждый сетевой адаптер принимает широковещательные передачи. Все драйверы Ethernet проверяют поле типа в принятом Ethernet-кадре и передают ARP-пакеты модулю ARP. ARP-запрос можно интерпретировать так: "Если ваш IP-адрес совпадает с указанным, то сообщите мне ваш Ethernet-адрес". Пакет ARP-запроса выглядит примерно так.

Таблица 3 - Пример ARP-запроса

IP-адрес отправителя	223.1.2.1
Ethernet-адрес отправителя	08:00:39:00:2F:C3
Искомый IP-адрес	223.1.2.2
Искомый Ethernet-адрес	<пусто>

Каждый модуль ARP проверяет поле искомого IP-адреса в полученном ARP-пакете и, если адрес совпадает с его собственным IP-адресом, то посылает ответ прямо по Ethernet-адресу отправителя запроса. ARP-ответ можно интерпретировать так: "Да, это мой IP-адрес, ему соответствует такой-то Ethernet-адрес". Пакет с ARP-ответом выглядит примерно так.

Таблица 4 - Пример ARP-ответа

IP-адрес отправителя	223.1.2.2
Ethernet-адрес отправителя	08:00:28:00:38:A9
Искомый IP-адрес	223.1.2.1
Искомый Ethernet-адрес	08:00:39:00:2F:C3

Этот ответ получает машина, сделавшая ARP-запрос. Драйвер этой машины проверяет поле типа в Ethernet-кадре и передает ARP-пакет модулю ARP. Модуль ARP анализирует

ARP-пакет и добавляет запись в свою ARP-таблицу. Обновленная таблица выглядит следующим образом.

Таблица 5 - ARP-таблица после обработки ответа

IP-адрес	Ethernet-адрес
223.1.2.1	08:00:39:00:2F:C3
223.1.2.2	08:00:28:00:38:A9
223.1.2.3	08:00:5A:21:A7:22
223.1.2.4	08:00:10:99:AC:54

Продолжение преобразования адресов.Новая запись в ARP-таблице появляется автоматически, спустя несколько миллисекунд после того, как она потребовалась. Как вы помните, ранее на шаге 2 исходящий IP-пакет был поставлен в очередь. Теперь с использованием обновленной ARP-таблицы выполняется преобразование IPадреса в Ethernet-адрес, после чего Ethernet-кадр передается по сети. Полностью порядок преобразования адресов выглядит так:

1) По сети передается широковещательный ARP-запрос.

2) Исходящий IP-пакет ставится в очередь.

3) Возвращается ARP-ответ, содержащий информацию о соответствии IP- и Ethernet-адресов. Эта информация заносится в ARP-таблицу.

4) Для преобразования IP-адреса в Ethernet-адрес у IP-пакета, поставленного в очередь, используется ARP-таблица.

5) Ethernet-кадр передается по сети Ethernet.

Короче говоря, если с помощью ARP-таблицы не удается сразу осуществить преобразование адресов, то IP-пакет ставится в очередь, а необходимая для преобразования информация получается с помощью запросов и ответов протокола ARP, после чего IP-пакет передается по назначению.

Если в сети нет машины с искомым IP-адресом, то ARP-ответа не будет и не будет записи в ARP-таблице. Протокол IP будет уничтожать IP-пакеты, направляемые по этому адресу. Протоколы верхнего уровня не могут отличить случай повреждения сети Ethernet от случая отсутствия машины с искомым IP-адресом.

Некоторые реализации IP и ARP не ставят в очередь IP-пакеты на то время, пока они ждут ARP-ответов. Вместо этого IP-пакет просто уничтожается, а его восстановление возлагается на модуль TCP или прикладной процесс, работающий через UDP. Такое восстановление выполняется с помощью таймаутов и повторных передач. Повторная

передача сообщения проходит успешно, так как первая попытка уже вызвала заполнение ARP-таблицы.

Следует отметить, что каждая машина имеет отдельную ARP-таблицу для каждого своего сетевого интерфейса.

IP-маршрутизация

Модуль IP является базовым элементом технологии Internet, а центральной частью IP является его таблица маршрутов. Протокол IP использует эту таблицу при принятии всех решений о маршрутизации IP-пакетов. Содержание таблицы маршрутов определяется администратором сети. Ошибки при установке маршрутов могут заблокировать передачи.

Чтобы понять технику межсетевого взаимодействия, нужно понять то, как используется таблица маршрутов. Это понимание необходимо для успешного администрирования и сопровождения IP-сетей.

Прямая маршрутизация.На рисунке показана небольшая IP-сеть, состоящая из 3 машин: A, B и C. Каждый сетевой адаптер этих машин имеет свой Ethernet-адрес. Менеджер сети должен присвоить машинам уникальные IP-адреса.

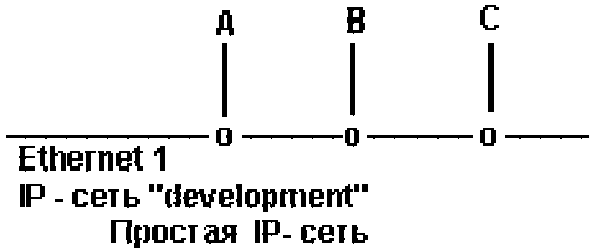


Рисунок 2 - Простая IP-сеть

Когда A посылает IP-пакет B, то заголовок IP-пакета содержит в поле отправителя IP-адрес узла A, а заголовок Ethernet-кадра содержит в поле отправителя Ethernet-адрес A. Кроме этого, IP-заголовок содержит в поле получателя IP-адрес узла B, а Ethernet-заголовок содержит в поле получателя Ethernet-адрес B.

Таблица 6 - Адреса в Ethernet-кадре, передающем IP-пакет от A к B

Адрес	Отправитель	Получатель
IP-заголовок	A	B
Ethernet-заголовок	A	B

В этом простом примере протокол IP является излишеством, которое мало что добавляет к услугам, предоставляемым сетью Ethernet. Однако протокол IP требует

дополнительных расходов на создание, передачу и обработку IP-заголовка. Когда в машине B модуль IP получает IP-пакет от машины A, он сопоставляет IP-адрес места назначения со своим, и если адреса совпадают, то передает дейтаграмму протоколу верхнего уровня.

В данном случае при взаимодействии A с B используется прямая маршрутизация.

Косвенная маршрутизация.На рисунке представлена более реалистичная картина сети Internet. В данном случае сеть Internet состоит из трех сетей Ethernet, на базе которых работают три IP-сети, объединенные шлюзом D. Каждая IP-сеть включает четыре машины; каждая машина имеет свои собственные IP- и Ethernet адреса.

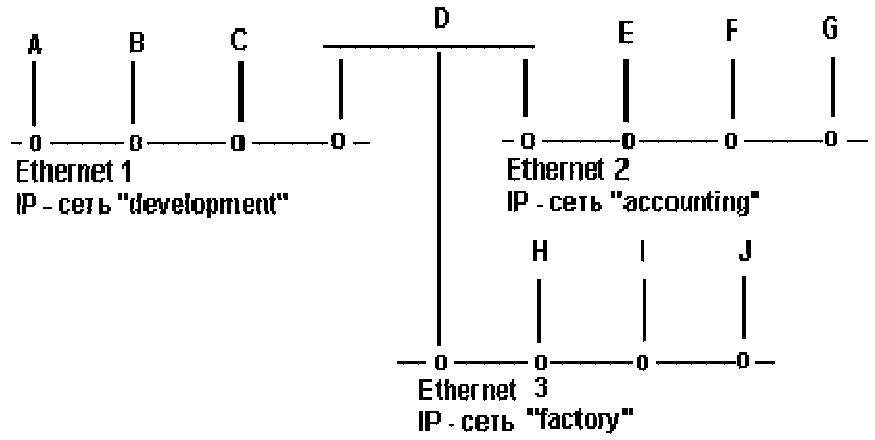


Рисунок 3 - Сеть Internet, состоящая из трех IP-сетей

Шлюз D соединяет все три сети и, следовательно, имеет три IP-адреса и три Ethernet-адреса. Машина D имеет три модуля ARP и три драйвера Ethernet. Обратим внимание на то, что машина D имеет только один модуль IP.

Менеджер сети присваивает каждой сети Ethernet уникальный номер, называемый IP-номером сети. На рисунке 15 IP-номера не показаны, вместо них используются имена сетей.

Когда машина A посылает IP-пакет машине B, то процесс передачи идет в пределах одной сети. При всех взаимодействиях между машинами, подключенными к одной IP-сети, используется прямая маршрутизация, обсуждавшаяся в предыдущем примере.

Когда машина D взаимодействует с машиной A, то это прямое взаимодействие. Когда машина D взаимодействует с машиной E, то это прямое взаимодействие. Когда машина D взаимодействует с машиной H, то это прямое взаимодействие. Это так, поскольку каждая пара этих машин принадлежит одной IP-сети.

Однако когда машина A взаимодействует с машинами, включенными в другую IP-сеть, то взаимодействие уже не будет прямым. Машина A должна использовать шлюз D для ретрансляции IP-пакетов в другую IP-сеть. Такое взаимодействие называется "косвенным".

Маршрутизация IP-пакетов выполняется модулями IP и является прозрачной для модулей TCP, UDP и прикладных процессов.

Если машина A посылает машине E IP-пакет, то IP-адрес и Ethernet-адрес отправителя соответствуют адресам A. IP-адрес места назначения является адресом E, но поскольку модуль IP в A посылает IP-пакет через D, Ethernet-адрес места назначения является адресом D.

Таблица 7 - Адреса в Ethernet-кадре, содержащем IP-пакет от A к E (до шлюза D).

Адрес	Отправитель	Получатель
IP-заголовок	A	E
Ethernet-заголовок	A	D

Модуль IP в машине D получает IP-пакет и проверяет IP-адрес места назначения. Определив, что это не его IP-адрес, шлюз D посылает этот IP-пакет прямо к E.

Таблица 8 - Адреса в Ethernet-кадре, содержащем IP-пакет от A к E (после шлюз D)

Адрес	Отправитель	Получатель
IP-заголовок	A	E
Ethernet-заголовок	D	E

Итак, при прямой маршрутизации IP- и Ethernet-адреса отправителя соответствуют адресам того узла, который послал IP-пакет, а IP- и Ethernet-адреса места назначения соответствуют адресам получателя. При косвенной маршрутизации IP- и Ethernet-адреса не образуют таких пар.

В данном примере сеть internet является очень простой. Реальные сети могут быть гораздо сложнее, так как могут содержать несколько шлюзов и несколько типов физических сред передачи. В приведенном примере несколько сетей Ethernet объединяются шлюзом для того, чтобы локализовать широковещательный трафик в каждой сети.

Правила маршрутизации в модуле IP.Рассмотрим правила или алгоритм маршрутизации. Для отправляемых IP-пакетов, поступающих от модулей верхнего уровня, модуль IP должен определить способ доставки - прямой или косвенный - и выбрать сетевой интерфейс. Этот выбор делается на основании результатов поиска в таблице маршрутов.

Для принимаемых IP-пакетов, поступающих от сетевых драйверов, модуль IP должен решить, нужно ли ретранслировать IP-пакет по другой сети или передать его на верхний уровень. Если модуль IP решит, что IP-пакет должен быть ретранслирован, то дальнейшая работа с ним осуществляется также, как с отправляемыми IP-пакетами.

Входящий IP-пакет никогда не ретранслируется через тот же сетевой интерфейс, через который он был принят.

Решение о маршрутизации принимается до того, как IP-пакет передается сетевому драйверу, и до того, как происходит обращение к ARP-таблице.

IP-таблица маршрутов. Как модуль IP узнает, какой именно сетевой интерфейс нужно использовать для отправления IP-пакета? Модуль IP осуществляет поиск в таблице маршрутов. Ключом поиска служит номер IP-сети, выделенный из IP-адреса места назначения IP-пакета.

Таблица маршрутов содержит по одной строке для каждого маршрута. Основными столбцами таблицы маршрутов являются номер сети, флаг прямой или косвенной маршрутизации, IP-адрес шлюза и номер сетевого интерфейса. Эта таблица используется модулем IP при обработке каждого отправляемого IP-пакета.

В большинстве систем таблица маршрутов может быть изменена с помощью команды "route". Содержание таблицы маршрутов определяется менеджером сети, поскольку менеджер сети присваивает машинам IP-адреса.

Подробности прямой маршрутизации. Рассмотрим более подробно, как происходит маршрутизация в одной физической сети.

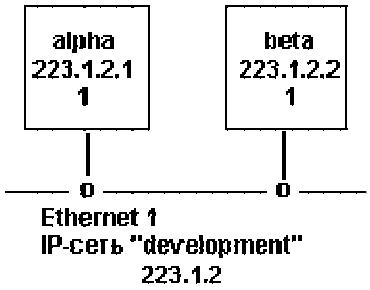


Рисунок 4 - Одна физическая сеть

Таблица маршрутов в узле alpha выглядит так.

Таблица 9 - Пример таблицы маршрутов

Сеть	Флаг вида маршрутизации	Шлюз	Номер интерфейса
development	Прямая	<пусто>	1

В данном простом примере все узлы сети имеют одинаковые таблицы маршрутов. Для сравнения ниже представлена та же таблица, но вместо названия сети указан ее номер.

Таблица 10 - Пример таблицы маршрутов с номерами сетей

Сеть	Флаг вида маршрутизации	Шлюз	Номер интерфейса
223.1.2	Прямая	<пусто>	1

Порядок прямой маршрутизации. Узел alpha посылает IP-пакет узлу beta. Этот пакет находится в модуле IP узла alpha, и IP-адрес места назначения равен IP-адресу beta (223.1.2.2). Модуль IP с помощью маски подсети выделяет номер сети из IP-адреса и ищет соответствующую ему строку в таблице маршрутов. В данном случае подходит первая строка.

Остальная информация в найденной строке указывает на то, что машины этой сети доступны напрямую через интерфейс номер 1. С помощью ARP-таблицы выполняется преобразование IP-адреса в соответствующий Ethernet-адрес, и через интерфейс 1 Ethernet-кадр посылается узлу beta.

Если прикладная программа пытается послать данные по IP-адресу, который не принадлежит сети development, то модуль IP не сможет найти соответствующую запись в таблице маршрутов. В этом случае модуль IP отбрасывает IP-пакет. Некоторые реализации протокола возвращают сообщение об ошибке "Сеть не доступна".

Подробности косвенной маршрутизации.Теперь рассмотрим более сложный порядок маршрутизации в IP-сети, изображенной на рисунке 17.

Таблица маршрутов в узле alpha выглядит так.

Таблица 11 - Таблица маршрутов в узле alpha

Сеть	Флаг вида маршрутизации	Шлюз	Номер интерфейса
development	прямая	<пусто>	1
accounting	косвенная	devnetrouter	1
factory	косвенная	devnetrouter	1

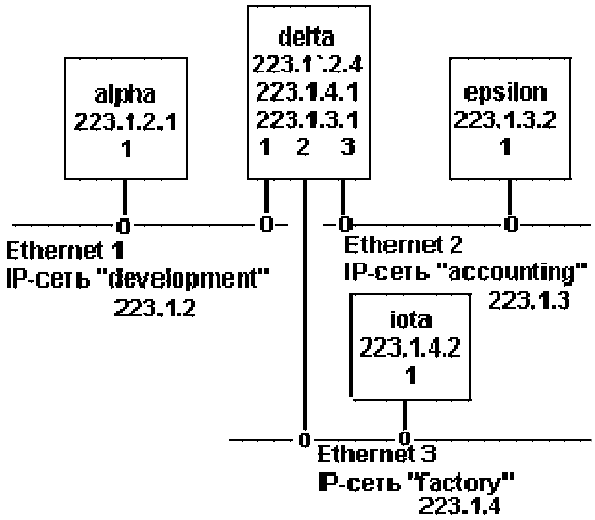


Рисунок 5 - Подробная схема трех сетей

Та же таблица с IP-адресами вместо названий.

Таблица 12 - Таблица маршрутов в узле alpha (с номерами)

Сеть	Флаг вида маршрутизации	Шлюз	Номер интерфейса
223.1.2	прямая	<пусто>	1
223.1.3	косвенная	223.1.2.4	1
223.1.4	косвенная	223.1.2.4	1

В столбце "шлюз" таблицы маршрутов узла alpha указывается IP-адрес точки соединения узла delta с сетью development.

Порядок косвенной маршрутизации.Узел alpha посылает IP-пакет узлу epsilon. Этот пакет находится в модуле IP узла alpha, и IP-адрес места назначения равен IP-адресу узла epsilon (223.1.3.2). Модуль IP выделяет сетевой номер из IP-адреса (223.1.3) и ищет соответствующую ему строку в таблице маршрутов. Соответствие находится во второй строке.

Запись в этой строке указывает на то, что машины требуемой сети доступны через шлюз devnetrouter. Модуль IP в узле alpha осуществляет поиск в ARP-таблице, с помощью которого определяет Ethernet-адрес, соответствующий IP-адресу devnetrouter. Затем IP-пакет, содержащий IP-адрес места назначения epsilon, посылается через интерфейс 1 шлюзу devnetrouter.

IP-пакет принимается сетевым интерфейсом в узле delta и передается модулю IP. Проверяется IP-адрес места назначения, и, поскольку он не соответствует ни одному из собственных IP-адресов delta, шлюз решает ретранслировать IP-пакет. Модуль IP в узле delta выделяет сетевой номер из IP-адреса места назначения IP-пакета (223.1.3) и ищет соответствующую запись в таблице маршрутов. Таблица маршрутов в узле delta выглядит так.

Таблица 13 - Таблица маршрутов в узле delta

Сеть	Флаг вида маршрутизации	Шлюз	Номер интерфейса
development	прямая	<пусто>	1
accounting	прямая	<пусто>	2
factory	прямая	<пусто>	3

Та же таблица с IP-адресами вместо названий.

Таблица 14 - Таблица маршрутов в узле delta (с номерами)

Сеть	Флаг вида маршрутизации	Шлюз	Номер интерфейса
223.1.2	прямая	<пусто>	1
223.1.3	прямая	<пусто>	2
223.1.4	прямая	<пусто>	3

Соответствие находится во второй строке. Теперь модуль IP напрямую посылает IP-пакет узлу epsilon через интерфейс номер 2. Пакет содержит IP- и Ethernet-адреса места назначения равные epsilon.

Узел epsilon принимает IP-пакет, и его модуль IP проверяет IP-адрес места назначения. Он соответствует IP-адресу epsilon, поэтому содержащееся в IP-пакете сообщение передается протокольному модулю верхнего уровня.

Формат заголовка IP-дейтаграммы

IP-дейтаграмма состоит из заголовка и данных. Заголовок дейтаграммы состоит из 32-разрядных слов и имеет переменную длину, зависящую от размера поля “Options”, но всегда кратную 32 битам. За заголовком непосредственно следуют данные, передаваемые в дейтаграмме (рисунок 6).

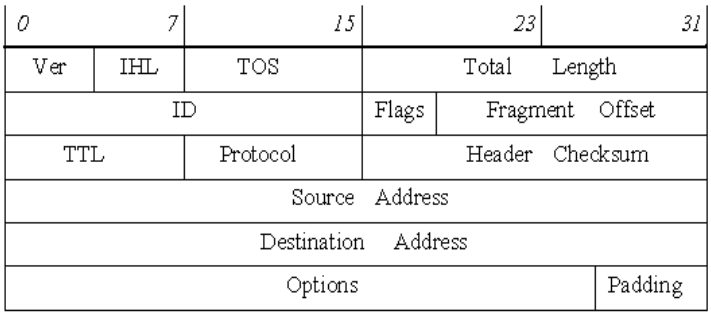


Рисунок 6 - Формат заголовка IP-дейтаграммы

Значения полей заголовка следующие.

Ver (4 бита) - версия протокола IP, в настоящий момент используется версия 4, новые разработки имеют номера версий 6-8.

IHL (Internet Header Length) (4 бита) - длина заголовка в 32-битных словах; диапазон допустимых значений от 5 (минимальная длина заголовка, поле “Options” отсутствует) до 15 (т.е. может быть максимум 40 байт опций).

TOS (Type Of Service) (8 бит) - значение поля определяет приоритет дейтаграммы и желаемый тип маршрутизации. Структура байта TOS представлена на рисунке 7.

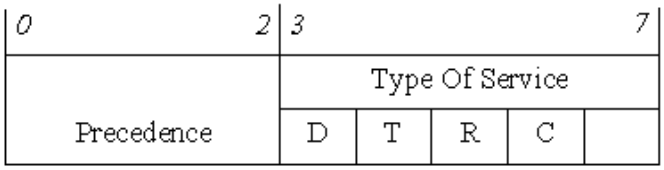


Рисунок 7 - Структура байта TOS

Три младших бита (“Precedence”) определяют приоритет дейтаграммы: 111 - управление сетью, 110 - межсетевое управление, 101 - CRITIC-ECP, 100 - более чем мгновенно, 011 – мгновенно, 010 – немедленно, 001 – срочно, 000 – обычно.

Биты D,T,R,C определяют желаемый тип маршрутизации:

- D (Delay) - выбор маршрута с минимальной задержкой,
- T (Throughput) - выбор маршрута с максимальной пропускной способностью,
- R (Reliability) - выбор маршрута с максимальной надежностью,
- C (Cost) - выбор маршрута с минимальной стоимостью.

В дейтаграмме может быть установлен только один из битов D,T,R,C. Старший бит байта не используется.

Реальный учет приоритетов и выбора маршрута в соответствии со значением байта TOS зависит от маршрутизатора, его программного обеспечения и настроек. Маршрутизатор может поддерживать расчет маршрутов для всех типов TOS, для части или игнорировать

TOS вообще. Маршрутизатор может учитывать значение приоритета при обработке всех дейтаграмм или при обработке дейтаграмм, исходящих только из некоторого ограниченного множества узлов сети, или вовсе игнорировать приоритет.

Total Length (16 бит) - длина всей дейтаграммы в октетах, включая заголовок и данные, максимальное значение 65535, минимальное - 21 (заголовок без опций и один октет в поле данных).

ID (Identification) (16 бит), **Flags** (3 бита), **Fragment Offset** (13 бит) используются для фрагментации и сборки дейтаграмм.

TTL (Time To Live) (8 бит) - "время жизни" дейтаграммы. Устанавливается отправителем, измеряется в секундах. Каждый маршрутизатор, через который проходит дейтаграмма, переписывает значение TTL, предварительно вычтя из него время, потраченное на обработку дейтаграммы. Так как в настоящее время скорость обработки данных на маршрутизаторах велика, на одну дейтаграмму тратится обычно меньше секунды, поэтому фактически каждый маршрутизатор вычитает из TTL единицу. При достижении значения TTL=0 дейтаграмма уничтожается, при этом отправителю может быть послано соответствующее ICMP-сообщение. Контроль TTL предотвращает зацикливание дейтаграммы в сети.

Protocol (8 бит) - определяет программу (вышестоящий протокол стека), которой должны быть переданы данные дейтаграммы для дальнейшей обработки.

Header Checksum (16 бит) - контрольная сумма заголовка, представляет из себя 16 бит, дополняющие биты в сумме всех 16-битовых слов заголовка. Перед вычислением контрольной суммы значение поля "Header Checksum" обнуляется. Поскольку маршрутизаторы изменяют значения некоторых полей заголовка при обработке дейтаграммы (как минимум, поля "TTL"), контрольная сумма каждым маршрутизатором пересчитывается заново. Если при проверке контрольной суммы обнаруживается ошибка, дейтаграмма уничтожается.

Source Address (32 бита) - IP-адрес отправителя.

Destination Address (32 бита) - IP-адресполучателя.

Options - опции, поле переменной длины. Опций может быть одна, несколько или ни одной. Опции определяют дополнительные услуги модуля IP по обработке дейтаграммы, в заголовок которой они включены.

Padding - выравнивание заголовка по границе 32-битного слова, если список опций занимает нецелое число 32-битных слов. Поле "Padding" заполняется нулями.

Протокол ICMP

Протокол ICMP(Internet Control Message Protocol, Протокол Управляющих Сообщений Интернет) выполняет следующие задачи:

- сообщает узлу-источнику об отказах маршрутизации;

- проверяет способности узлов образовывать повторное эхо в объединенной сети (сообщения Echo и ReplyICMP);

- стимулирует более эффективную маршрутизацию (с помощью сообщений RedirectICMP - переадресации ICMP);

- информирует узел-источник о том, что некоторая дейтаграмма превысила назначенное ей время существования в пределах данной сети (сообщение TimeExceededICMP - "время превышено");

- обеспечивает для новых узлов возможность нахождения маски подсети, используемой в объединенной сети в данный момент.

Протокол ICMP является неотъемлемой частью IP-модуля. Он обеспечивает обратную связь в виде диагностических сообщений, посылаемых отправителю при невозможности доставки его дейтаграммы и в других случаях.

ICMP-сообщения не порождаются при невозможности доставки:

– дейтаграмм, содержащих ICMP-сообщения;

– не первых фрагментов дейтаграмм;

– дейтаграмм, направленных по групповому адресу (широковещание, мультикастинг);

– дейтаграмм, адрес отправителя которых нулевой или групповой. Все ICMP-сообщения имеют IP-заголовок, значение поля “Protocol” равно 1.

Данные дейтаграммы с ICMP-сообщением не передаются вверх по стеку протоколов для обработки, а обрабатываются IP-модулем.

После IP-заголовка следует 32-битное слово с полями “Тип”, “Код” и “Контрольная сумма”. Поля типа и кода определяют содержание ICMP-сообщения. Формат остальной части дейтаграммы зависит от вида сообщения.

Контрольная сумма считается так же, как и в IP-заголовке, но в этом случае суммируется содержимое ICMP-сообщения, включая поля “Тип” и “Код”.

Протокол дейтаграмм пользователя UDP

Протокол UDP (User Datagram Protocol, протокол пользовательских дейтаграмм) используется в тех случаях, когда мощные средства обеспечения надежности протокола TCP не требуются. Протокол UDP обеспечивает ненадежную доставку дейтаграмм и не поддерживает соединений из конца в конец. К заголовку IP-пакета он добавляет два поля, одно из которых, поле "порт", обеспечивает мультиплексирование информации между разными прикладными процессами, а другое поле - "контрольная сумма" - позволяет поддерживать целостность данных. Реализация UDP намного проще, чем TCP.

Протокол UDP используется либо при пересылке коротких сообщений, когда накладные расходы на установление сеанса и проверку успешной доставки данных оказываются выше расходов на повторную (в случае неудачи) пересылку сообщения, либо в том случае, когда сама организация процесса-приложения обеспечивает установление соединения и проверку доставки пакетов (например, NFS).

Пользовательские данные, поступившие от прикладного уровня, предваряются UDP-заголовком, и сформированный таким образом UDP-пакет отправляется на межсетевой уровень. UDP-заголовок состоит из двух 32-битных слов (рисунок 8).

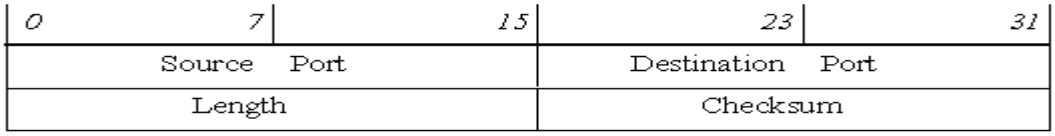


Рисунок 8 - UDP-заголовок

Заголовок UDP имеет четыре поля:

- порт источника (sourceport) - те же функции, что и в заголовке TCP;
- порт пункта назначения (destinationport) - те же функции, что и в заголовке TCP;
- длина (length) - длина заголовка UDP и данных;
- контрольная сумма (checksum) - обеспечивает проверку целостности пакета (факультативная возможность).

Контрольное суммирование.Контрольная сумма вычисляется таким же образом, как и в TCP-заголовке.Когда модуль UDP получает дейтаграмму от модуля IP, он проверяет контрольную сумму, содержащуюся в ее заголовке. Если контрольная сумма равна нулю, то это означает, что отправитель дейтаграммы ее не подсчитывал, и, следовательно, ее нужно игнорировать. Если два модуля UDP взаимодействуют только через одну сеть Ethernet, то от контрольного суммирования можно отказаться, так как средства Ethernet обеспечивают достаточную степень надежности обнаружения ошибок передачи. Это снижает накладные расходы, связанные с работой UDP. Однако рекомендуется всегда выполнять контрольное суммирование, так как возможно в какой-то момент изменения в таблице маршрутов приведут к тому, что дейтаграммы будут посылаться через менее надежную среду.

Если контрольная сумма правильная, то проверяется порт назначения, указанный в заголовке дейтаграммы. Если к этому порту подключен прикладной процесс, то прикладное сообщение, содержащееся в дейтаграмме, становится в очередь для прочтения. В остальных случаях дейтаграмма отбрасывается. Если дейтаграммы поступают быстрее, чем их успевает обрабатывать прикладной процесс, то при переполнении очереди сообщений поступающие дейтаграммы отбрасываются модулем UDP.

После заголовка непосредственно следуют пользовательские данные, переданные модулю UDP прикладным уровнем за один вызов. Протокол UDP рассматривает эти данные как целостное сообщение; он никогда не разбивает сообщение для передачи в нескольких пакетах и не объединяет несколько сообщений для пересылки в одном пакете. Если прикладной процесс N раз вызвал модуль UDP для отправки данных (т.е. запросил отправку N сообщений), то модулем UDP будет сформировано и отправлено N пакетов, и процесс-получатель будет должен N раз вызвать свой модуль UDP для получения всех сообщений.

При получении пакета от межсетевого уровня модуль UDP проверяет контрольную сумму и передает содержимое сообщения прикладному процессу, чей номер порта указан в поле “Destination Port”.

Максимальная длина UDP-сообщения равна максимальной длине IP-дейтаграммы (65535 октетов) за вычетом минимального IP-заголовка (20) и UDP-заголовка (8), т.е. 65507 октетов. На практике обычно используются сообщения длиной 8192 октета.

Примеры прикладных процессов, использующих протокол UDP: NFS (Network File System - сетевая файловая система), TFTP (Trivial File Transfer Protocol - простой протокол передачи файлов), SNMP (Simple Network Management Protocol - простой протокол управления сетью), DNS (Domain Name Service - доменная служба имен).

Порты. Взаимодействие между прикладными процессами и модулем UDP осуществляется через UDP-порты. Порты нумеруются, начиная с нуля. Прикладной процесс, предоставляющий некоторые услуги другим прикладным процессам (сервер), ожидает поступления сообщений в порт, специально выделенный для этих услуг. Сообщения должны содержать запросы на предоставление услуг. Они отправляются процессами-клиентами.

Например, сервер SNMP всегда ожидает поступлений сообщений в порт 161. Если клиент SNMP желает получить услугу, он посылает запрос в UDP порт 161 на машину, где работает сервер. В каждом узле может быть только один сервер SNMP, так как существует только один UDP-порт 161. Данный номер порта является общеизвестным, то есть фиксированным номером, официально выделенным для услуг SNMP. Общеизвестные номера определяются стандартами Internet.

По номеру порта транспортные протоколы определяют, какому приложению передать содержимое пакетов.

2.6 Лабораторная работа № 6 (2 часа)

Тема: «Кодирование информации»

2.6.1 Цель работы: изучить способы кодирования информации в вычислительных сетях.

2.6.2 Задачи работы:

1. рассмотреть методы физического кодирования;
2. ознакомиться с методами повышения помехоустойчивости передачи и приема;
3. разработать программу для кодирования информации.

2.6.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. Компьютерная ЛВС.

2.6.4 Описание (ход) работы:

1. Методы физического кодирования

При выборе способа кодирования нужно одновременно стремиться к достижению нескольких целей:

- минимизировать ширину спектра сигнала, полученного в результате кодирования;
- обеспечивать синхронизацию между передатчиком и приемником;
- обеспечивать устойчивость к шумам;
- обнаруживать и по возможности исправлять битовые ошибки;
- минимизировать мощность передатчика.

Более узкий спектр сигнала позволяет на одной и той же линии (с одной и той же полосой пропускания) добиваться более высокой скорости передачи данных. Спектр сигнала в общем случае зависит как от способа кодирования, так и от тактовой частоты передатчика.

Требования, предъявляемые к методам кодирования, являются взаимно противоречивыми, поэтому каждый из рассматриваемых далее популярных методов кодирования обладает своими достоинствами и недостатками в сравнении с другими.

Метод биполярного кодирования с альтернативной инверсией (AMI)

Одной из модификаций метода NRZ является метод биполярного кодирования с альтернативной инверсией (Bipolar Alternate Mark Inversion, AMI). В этом методе используются три уровня потенциала — отрицательный, нулевой и положительный. Для кодирования логического нуля используется нулевой потенциал, а логическая единица кодируется либо положительным потенциалом, либо отрицательным, при этом потенциал каждой новой единицы противоположен потенциалу предыдущей.

Код AMI частично ликвидирует проблемы постоянной составляющей и отсутствия самосинхронизации, присущие коду NRZ. Это происходит при передаче длинных последовательностей единиц. В этих случаях сигнал на линии представляет собой последовательность разнополярных импульсов с тем же спектром, что и у кода NRZ,

передающего чередующиеся нули и единицы, то есть без постоянной составляющей и с основной гармоникой N/2 Гц (где N — битовая скорость передачи данных). Длинные же последовательности нулей также опасны для кода AMI, как и для кода NRZ — сигнал вырождается в постоянный потенциал нулевой амплитуды. Поэтому код AMI требует дальнейшего улучшения.

Потенциальный код с инверсией при единице (NRZI)

Существует код, похожий на AMI, но только с двумя уровнями сигнала. При передаче нуля он передает потенциал, который был установлен в предыдущем такте (то есть не меняет его), а при передаче единицы потенциал инвертируется на противоположный. Этот код называется потенциальным кодом с инверсией при единице (Non Return to Zero with ones Inverted, NRZI). Этот код удобен в тех случаях, когда использование третьего уровня сигнала весьма нежелательно, например в оптических кабелях, где устойчиво распознаются два состояния сигнала - свет и темнота. Для улучшения потенциальных кодов, подобных AMI и NRZI, используются два метода. Первый метод основан на добавлении в исходный код избыточных бит, содержащих логические единицы. В этом случае длинные последовательности 0-ей прерываются, и код становится самосинхронизирующимся для любых передаваемых данных. Исчезает также постоянная составляющая, а значит, еще более сужается спектр сигнала. Но этот метод снижает полезную пропускную способность линии, так как избыточные единицы пользовательской информации не несут.

Манчестерский код

Манчестерский код (или код Манчестер-II) получил наибольшее распространение в локальных сетях. Он также относится к самосинхронизирующимся кодам, имеет два уровня, что способствует его лучшей помехозащищенности и упрощению приемных и передающих узлов. Логическому нулю соответствует положительный переход в центре битового интервала (то есть первая половина битового интервала – низкий уровень, вторая половина – высокий), а логической единице соответствует отрицательный переход в центре битового интервала (или наоборот).

Обязательное наличие перехода в центре бита позволяет приемнику манчестерского кода легко выделить из пришедшего сигнала синхросигнал и передать информацию сколь угодно большими последовательностями без потерь из-за рассинхронизации.

Потенциальный код 2B1Q

Код 2B1Q его название отражает суть — каждые два бита (2B) передаются за один такт (1) сигналом, имеющим четыре состояния (Q — Quadra). Паре битов 00 соответствует потенциал -2,5 В, паре 01 — потенциал -0,833 В, паре 11 — потенциал +0,833 В, а паре 10 — потенциал +2,5 В.

При этом способе кодирования требуются дополнительные меры по борьбе с длинными последовательностями одинаковых пар битов, так как при этом сигнал превращается в постоянную составляющую. При случайном чередовании битов спектр сигнала в два раза уже, чем у кода NRZ, так как при той же битовой скорости длительность такта увеличивается в два раза. Таким образом, с помощью кода 2B1Q можно по одной и той же линии передавать данные в два раза быстрее, чем с помощью кода AMI или NRZI.

Однако для его реализации мощность передатчика должна быть выше, чтобы четыре уровня четко различались приемником на фоне помех. Для улучшения потенциальных кодов типа AMI, NRZI или 2Q1B используются избыточные коды и скрэмблирование.

2. Методы повышения помехоустойчивости передачи и приема.

Логическое кодирование используется для улучшения потенциальных кодов типа AMI, NRZI, 2Q1B и уменьшения помех в сети. Логическое кодирование должно заменять длинные последовательности бит, приводящие к постоянному потенциалу (перегрев оборудования), вкраплениями единиц. Для логического кодирования характерны два метода - избыточные коды и скрэмблирование.

Избыточные коды

Избыточные коды основаны на разбиении исходной последовательности битов на порции, которые часто называют символами. Затем каждый исходный символ заменяется новым с большим количество битов, чем исходный.

Например, в логическом **коде 4B/5B**, используемом в технологиях FDDI и FastEthernet, исходные символы длиной 4 бит заменяются символами длиной 5 бит. Так как результирующие символы содержат избыточные биты, то общее количество битовых комбинаций в них больше, чем в исходных.

Таблица 1. Соответствие исходных и результирующих кодов 4B/5B

Исходный код	Результирующий код	Исходный код	Результирующий код
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
0111	01111	1111	11101

Скремблирование

Методы скремблирования заключаются в побитном вычислении результирующего кода на основании бит исходного кода и полученных в предыдущих тактах бит результирующего кода. Например, скремблер может реализовывать следующее соотношение:

$B_i = A_i \oplus B_{i-3} \oplus B_{i-5}$,

где B_i — двоичная цифра результирующего кода, полученная на i-м такте работы скремблера,A_i — двоичная цифра исходного кода, поступающая на i-м такте на

вход скремблера, B_{i-3} и B_{i-5} — двоичные цифры результирующего кода, полученные на предыдущих тактах работы скремблера, соответственно на 3 и на 5 тактов ранее текущего такта, ⊕— операция исключающего ИЛИ (сложение по модулю 2).

Например, для исходной последовательности 110110000001 скремблер даст следующий результирующий код:

$B_1 = A_1 = 1$ (первые три цифры результирующего кода будут совпадать с исходным, так как еще нет нужных предыдущих цифр)

$B_2 = A_2 = 1$

$B_3 = A_3 = 0$

$B_4 = A_4 \oplus B_1 = 1 \oplus 1 = 0$

$B_5 = A_5 \oplus B_2 = 1 \oplus 1 = 0$

$B_6 = A_6 \oplus B_3 \oplus B_1 = 0 \oplus 0 \oplus 1 = 1$

$B_7 = A_7 \oplus B_4 \oplus B_2 = 0 \oplus 0 \oplus 1 = 1$

$B_8 = A_8 \oplus B_5 \oplus B_3 = 0 \oplus 0 \oplus 0 = 0$

$B_9 = A_9 \oplus B_6 \oplus B_4 = 0 \oplus 1 \oplus 0 = 1$

$B_{10} = A_{10} \oplus B_7 \oplus B_5 = 0 \oplus 1 \oplus 0 = 1$

$B_{11} = A_{11} \oplus B_8 \oplus B_6 = 0 \oplus 0 \oplus 1 = 1$

$B_{12} = A_{12} \oplus B_9 \oplus B_7 = 1 \oplus 1 \oplus 1 = 1$

Таким образом, на выходе скремблера появится последовательность 110001101111, в которой нет последовательности из шести нулей, присутствовавшей в исходном коде.

После получения результирующей последовательности приемник передает ее дескремблеру, который восстанавливает исходную последовательность на основании обратного соотношения:

$C_i = B_i \oplus B_{i-3} \oplus B_{i-5} = (A_i \oplus B_{i-3} \oplus B_{i-5}) \oplus B_{i-3} \oplus B_{i-5} = A_i$.

Существуют и более простые методы борьбы с последовательностями единиц, также относимые к классу скрэмблирования.

Для улучшения кода AMI используются два метода, основанные на искусственном искажении последовательности нулей запрещенными символами.

На рисунке 2 показано использование метода B8ZS (Bipolarwith 8-Zeros Substitution) и метода HDB3 (High-DensityBipolar 3-Zeros) для корректировки кода AMI. Исходный код состоит из двух длинных последовательностей нулей: в первом случае - из 8, а во втором - из 5.

Рис. 2 Коды B8ZS и HDB3. V - сигнал единицы запрещенной полярности; 1*-сигнал единицы корректной полярности, но заменившей 0 в исходном коде.

Код B8ZS исправляет только последовательности, состоящие из 8 нулей. Для этого он после первых трех нулей вместо оставшихся пяти нулей вставляет пять цифр: V-1*-0-V-1*. V здесь обозначает сигнал единицы, запрещенной для данного такта полярности, то есть сигнал, не изменяющий полярность предыдущей единицы, 1* - сигнал единицы корректной полярности, а знак звездочки отмечает тот факт, что в исходном коде в этом такте была не единица, а ноль. В результате на 8 тактах приемник наблюдает 2 искажения - очень маловероятно, что это случилось из-за шума на линии или других сбоев передачи. Поэтому приемник считает такие нарушения кодировкой 8 последовательных нулей и после приема заменяет их на исходные 8 нулей. Код B8ZS построен так, что его постоянная составляющая равна нулю при любых последовательностях двоичных цифр.

Код HDB3 исправляет любые четыре подряд идущих нуля в исходной последовательности. Правила формирования кода: каждые четыре нуля заменяются четырьмя сигналами, в которых имеется один сигнал V. Для подавления постоянной составляющей полярность сигнала V чередуется при последовательных заменах. Для замены используются два образца четырехтактовых кодов. Если перед заменой исходный код содержал нечетное число 1-ц, то используется последовательность 000V, а если число 1-ц было четным - последовательность 1*00V.

Улучшенные потенциальные коды обладают достаточно узкой полосой пропускания для любых последовательностей единиц и нулей, которые встречаются в передаваемых данных.

3. Задание

Необходимо разработать программу для кодирования информации, используя код (по варианту), при этом для устранения последовательностей нулей использовать логическое кодирование (по варианту). Входную последовательность информации ввести с клавиатуры. Результаты работы отобразить в виде временной диаграммы, при этом на диаграмме должны быть:

- входная последовательность в коде NRZ,
- входная последовательность в виде самосинхронизирующегося кода (по варианту),
- входная последовательность в логическом коде (по варианту).

4. Варианты задания:

№ варианта	Самосинхронизирующиеся коды	Логическое кодирование
1	Биполярный код AMI	Избыточный код 4B/5B
2	Код NRZI	Скремблер со сдвигом 3 и 5
3	Манчестерский код	Избыточный код 4B/5B
4	Биполярный код AMI	Метод B8ZS
5	Код NRZI	Скремблер со сдвигами 3 и 5 позиции
6	Биполярный код AMI	Метод HDB3
7	Манчестерский код	Метод B8ZS
8	Код NRZI	Скремблер со сдвигом 5 и 8
9	Биполярный код AMI	Скремблер со сдвигом 3 и 5
10	Манчестерский код	Скремблер со сдвигом 3 и 5
11	Код NRZI	Метод B8ZS
12	Биполярный код AMI	Скремблер со сдвигами 5 и 13 позиции
13	Код NRZI	Метод HDB3
14	Код 2B1Q	Избыточный код 4B/5B
15	Код 2B1Q	Скремблер со сдвигом 3 и 5
16	Код 2B1Q	Метод B8ZS
17	Манчестерский код	Скремблер со сдвигами 5 и 13 позиции

5. Структура отчета

1. титульный лист;
2. цель работы, задание;
3. краткие теоретические сведения;
4. алгоритм работы программы;
5. листинг программы кодирования информации;
6. результаты работы программы;
7. выводы.

2.7 Лабораторная работа № 7 (2 часа)

Тема: «Разновидности архитектуры сетей»

2.7.1 Цель работы: изучить архитектуры локальных сетей.

2.7.2 Задачи работы:

1. рассмотреть архитектуру терминал-главный компьютер;

2. рассмотреть одноранговая архитектуру;

3. рассмотреть архитектуру клиент-сервер

2.7.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. Компьютерная ЛВС.

2.7.4 Описание (ход) работы:

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

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

Далее будет рассмотрено три вида архитектур:

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

Архитектура терминал-главный компьютер

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

Рассматриваемая архитектура предполагает два типа оборудования:

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

Главный компьютер через МПД взаимодействуют с терминалами, как представлено на рис. 1.

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

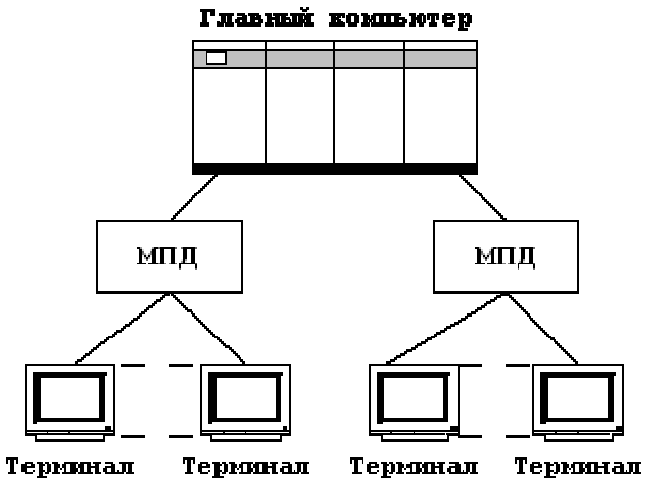


Рис. 1. Архитектура терминал-главный компьютер

Одноранговая архитектура

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

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

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

Одноранговые сети имеют следующие преимущества:

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

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

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

Архитектура клиент-сервер

Архитектура клиент-сервер (client-server architecture) – это концепция информационной сети, в которой основная часть ее ресурсов сосредоточена в серверах, обслуживающих своих клиентов (рис. 2). Рассматриваемая архитектура определяет два типа компонентов: серверы и клиенты.

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

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

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

Рис. 2. Архитектура клиент – сервер

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

Клиенты – это рабочие станции, которые используют ресурсы сервера и предоставляют удобные интерфейсы пользователя. Интерфейсы пользователя (рис. 3) это процедуры взаимодействия пользователя с системой или сетью.

В сетях с выделенным файловым сервером на выделенном автономном ПК устанавливается серверная сетевая операционная система. Этот ПК становится сервером. ПО, установленное на рабочей станции, позволяет ей обмениваться данными с сервером. Наиболее распространенные сетевые операционная системы:

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

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

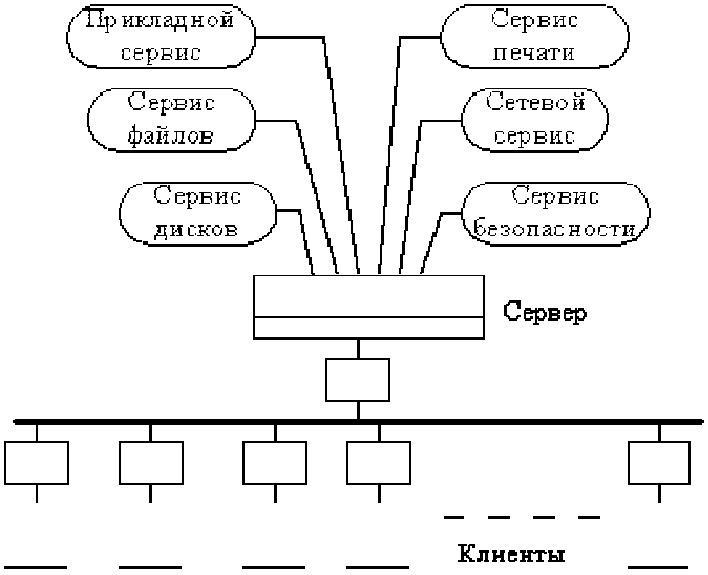


Рис. 3 – Модель клиент-сервер

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

1. Файл-серверы и принт-серверы. Они управляют доступом пользователей к файлам и принтерам. Так, например, для работы с текстовым документом вы прежде всего запускаете на своем компьютере (PC) текстовый процессор. Далее требуемый документ текстового процессора, хранящийся на файл-сервере, загружается в память PC, и таким образом Вы можете работать с этим документом на PC. Другими словами, файл-сервер предназначен для хранения файлов и данных.

2. Серверы приложений (в том числе сервер баз данных (БД), WEB-сервер). На них выполняются прикладные части клиент серверных приложений (программ). Эти серверы принципиально отличаются от файл-серверов тем, что при работе с файл-сервером нужный файл или данные целиком копируются на запрашивающий PC, а при работе с сервером приложений на PC пересылаются только результаты запроса. Например, по запросу можно получить только список работников, родившихся в сентябре, не загружая при этом в свою PC всю базу данных персонала.

3. Почтовые серверы управляют передачей электронных сообщений между пользователями сети.

4. Факс-серверы управляют потоком входящих и исходящих факсимильных сообщений через один или несколько факс-модемов.

5. Коммуникационные серверы управляют потоком данных и почтовых сообщений между данной ЛВС и другими сетями или удаленными пользователями через модем и телефонную линию. Они же обеспечивают доступ к Internet.

6. Сервер служб каталогов предназначен для поиска, хранения и защиты информации в сети. Windows NT Server объединяет PC в логические группы-домены, система защиты которых наделяет пользователей различными правами доступа к любому сетевому ресурсу.

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

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

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

Сети клиент-серверной архитектуры имеют следующие преимущества:

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

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

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

Выбор архитектуры сети

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

Следует выбрать одноранговую сеть, если:

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

Следует выбрать клиент-серверную сеть, если:

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