Министерство образования Республики Беларусь
Учреждение образования
«Гомельский государственный технический университет
имени П.О.Сухого»

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

КОНСПЕКТ ЛЕКЦИЙ
по дисциплине
«КОМПЬЮТЕРНЫЕ СЕТИ»
для студентов специальности
1-40 05 01 «Информационные системы и технологии (по направлениям)»
направление специальности 1–40 05 01–01 «Информационные системы и технологии (в проектировании и производстве)»

Курочка К.С., Соболев Д.В.

Гомель 2017

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Изучение дисциплины «Компьютерные сети» предусматривает получение студентами базовых знаний и основных умений применения и работы с компьютерными сетями.

Цель дисциплины – изучение теоретических основ принципов организации, проектирования, построения и использования вычислительных сетей, сетевых протоколов и их применение для организации взаимодействия объектов сети, основ передачи данных и базовых аппаратных средств для передачи информации, базовых технологий локальных сетей и принципов межсетевого взаимодействия.

Задачи дисциплины – получение практических навыков разработки сетевых, распределённых и многоуровневых приложений; подготовка специалиста, имеющего устойчивые навыки использования локальных и глобальных компьютерных сетей; формирование базовых навыков проектирования компьютерных сетей, эффективного использования и настройки сетевого оборудования; формирование навыков программирования сетевых технологий.

В результате изучения дисциплины обучаемый должен:

знать:

– основные концепциями построения локальных и глобальных сетей; методы объединения компьютеров и устройств в сети;

– основные функции и режимы взаимодействия компьютеров. Аппаратное и программное обеспечение сети;

– основные протоколы, методы организации, способы объединения компьютеров в сети;

– виды топологий сети и основные реализуемые алгоритмы взаимодействия узлов;

– способы передачи, методы кодирования и защиты данных;

– принципы разработки программ организации клиент-серверного взаимодействия. Методы разработки программ распределенной обработки данных;

– перспективные направления развития в области компьютерных сетей и сетевых технологий. Методы использования сетей и сетевых технологий в будущей профессиональной деятельности;

уметь:

– анализировать уровень эффективности сетевых решений;

– эффективно использовать операционные системы и предлагать сетевые решения для разрабатываемых прикладных задач;

– разрабатывать программы взаимодействия для работы в архитектуре клиент сервер для организации клиент-серверного взаимодействия и распределенной обработки данных;

– использовать различные протоколы при разработке программных средств.

владеть:

– навыками работы в сети и её администрирования;

– техникой работы с основными видами сетевого оборудования;

– методами создания правил маршрутизации и конфигурации интерфейсов сетевого оборудования;

– техникой и методами работы с сетевым окружением в различных операционных системах;

– технологией проектирования и разработки сетевого программного обеспечения.

Методика преподавания дисциплины «Компьютерные сети» строится на сочетании лекций и лабораторных занятий, проверки полученных знаний и самостоятельной работы.

Основными методами обучения, отвечающими целям изучения дисциплины, являются:

– элементы проблемного обучения (проблемное изложение), реализуемое на лекционных занятиях;

– элементы учебно-исследовательской деятельности, реализуемые на лабораторных занятиях и при самостоятельной работе;

– коммуникативные технологии (дискуссии, учебные дебаты), реализуемые на практических занятиях и конференциях.

Согласно учебному плану программа дисциплины «Компьютерные сети» рассчитана на объём 150 учебных часов: 68 аудиторных часов по дневной форме получения образования, 14 – по заочной, 8 – по заочной сокращенной. Трудоемкость учебной дисциплины – 4 зачетные единицы. Форма обучения – дневная, заочная, заочная сокращенная.

РАЗДЕЛ 2. ОБЩИЕ ПРИНЦИПЫ ПОСТРОЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ

Тема 2. Компьютерные сети. Основные понятия

Компьютерные сети являются логическим результатом эволюции развития компьютерных технологий. Постоянно возрастающие потребности пользователей в вычислительных ресурсах обуславливали попытки специалистов компьютерных технологий объединить в единую систему отдельные компьютеры.

Сеть – это соединение между двумя и более компьютерами, позволяющее им разделять ресурсы.

Сеть – это совокупность объектов, образуемых устройствами передачи и обработки данных. Международная организация по стандартизации определила вычислительную сеть *как последовательную бит-ориентированную передачу информации между связанными друг с другом независимыми устройствами.*

Сети обычно находится в частном ведении пользователя и занимают некоторую территорию и по территориальному признаку разделяются на:

- локальные вычислительные сети (ЛВС) или Local Area Network (LAN), расположенные в одном или нескольких близко расположенных зданиях. ЛВС обычно размещаются в рамках какой-либо организации (корпорации, учреждения), поэтому их называют корпоративными.
- распределенные компьютерные сети, глобальные или Wide Area Network (WAN), расположенные в разных зданиях, городах и странах, которые бывают территориальными, смешанными и глобальными. В зависимости от этого глобальные сети бывают четырех основных видов: городские, региональные, национальные и транснациональные. В качестве примеров распределенных сетей очень большого масштаба можно назвать: Internet, EUNET, Relcom, FIDO.

В состав сети в общем случае включается следующие элементы:

- сетевые компьютеры (оснащенные сетевым адаптером);
- каналы связи (кабельные, спутниковые, телефонные, цифровые, волоконно-оптические, радиоканалы и др.);
- различного рода преобразователи сигналов;
- сетевое оборудование.

Различают два понятия сети: *коммуникационная сеть* и *информационная сеть* (рис. 1).

Коммуникационная сеть предназначена для передачи данных, также она выполняет задачи, связанные с преобразованием данных. Коммуникационные сети различаются по типу используемых физических средств соединения.

Информационная сеть предназначена для хранения информации и состоит из *информационных систем*. На базе коммуникационной сети может быть построена группа информационных сетей:

Под *информационной системой* следует понимать систему, которая является поставщиком или потребителем информации.

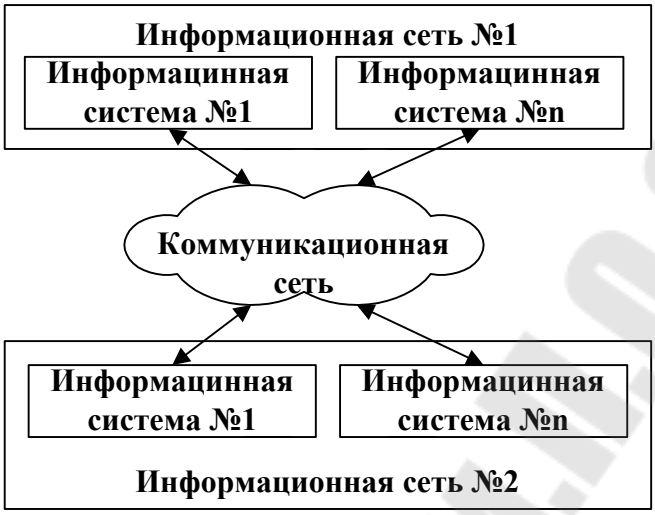


Рисунок 1 – Информационные и коммуникационные сети

Компьютерная сеть состоит из *информационных систем* и *каналов связи*.

Под *информационной системой* следует понимать объект, способный осуществлять хранение, обработку или передачу информация. В состав *информационной системы* входят: компьютеры, программы, пользователи и другие составляющие, предназначенные для процесса обработки и передачи данных. В дальнейшем информационная система, предназначенная для решения задач пользователя, будет называться – *рабочая станция (client)*. Рабочая станция в сети отличается от обычного персонального компьютера (ПК) наличием *сетевой карты* (*сетевого адаптера*), канала для передачи данных и сетевого программного обеспечения.

Под *каналом связи* следует понимать путь или средство, по которому передаются сигналы. Средство передачи сигналов называют *абонентским,* или *физическим, каналом*.

Каналы связи (data link) создаются по линиям связи при помощи сетевого оборудования и физических средств связи. Физические средства связи построены на основе витых пар, коаксиальных кабелей, оптических каналов или эфира. Между взаимодействующими информационными системами через физические каналы коммуникационной сети и узлы коммутации устанавливаются *логические каналы.*

Логический канал – это путь для передачи данных от одной системы к другой. Логический канал прокладывается по маршруту в одном или нескольких физических каналах. *Логический канал* можно охарактеризовать, как маршрут, проложенный через физические каналы и узлы коммутации.

Информация в сети передается *блоками данных* по процедурам обмена между объектами. Эти процедуры называют *протоколами передачи данных.*

Протокол – это совокупность правил, устанавливающих формат и процедуры обмена информацией между двумя или несколькими устройствами.

Загрузка сети характеризуется параметром, называемым *трафиком*. *Трафик (traffic)* – это поток сообщений в сети передачи данных. Под ним понимают количественное измерение в выбранных точках сети числа проходящих *блоков* данных и их длины, выраженное в битах в секунду.

Существенное влияние на характеристику сети оказывает метод доступа. Метод доступа – это способ определения того, какая из рабочих станций сможет следующей использовать канал связи и как управлять доступом к каналу связи (кабелю).

В сети все рабочие станции физически соединены между собою каналами связи по определенной структуре, называемой топологией. Топология – это описание физических соединений в сети, указывающее какие рабочие станции могут связываться между собой. Тип топологии определяет производительность, работоспособность и надежность эксплуатации рабочих станций, а также время обращения к файловому серверу. В зависимости от топологии сети используется тот или иной метод доступа.

Состав основных элементов в сети зависит от ее архитектуры. Архитектура – это концепция, определяющая взаимосвязь, структуру и функции взаимодействия рабочих станций в сети. Она предусматривает логическую, функциональную и физическую организацию технических и программных средств сети. Архитектура определяет принципы построения и функционирования аппаратного и программного обеспечения элементов сети.

В основном выделяют три вида архитектур: архитектура терминал – главный компьютер, архитектура клиент – сервер и одноранговая архитектура.

Современные сети можно классифицировать по различным признакам: по удаленности компьютеров, топологии, назначению, перечню предоставляемых услуг, принципам управления (централизованные и децентрализованные), методам коммутации, методам доступа, видам среды передачи, скоростям передачи данных и т. д. Все эти понятия будут рассмотрены более подробно при дальнейшем изучении курса.

Преимущества использования сетей.

Компьютерные сети представляют собой вариант сотрудничества людей и компьютеров, обеспечивающего ускорение доставки и обработки информации. Объединять компьютеры в сети начали более 30 лет назад. Когда возможности компьютеров выросли и ПК стали доступны каждому, развитие сетей значительно ускорилось.

Соединенные в сеть компьютеры обмениваются информацией и совместно используют периферийное оборудование и устройства хранения информации рис. 2.

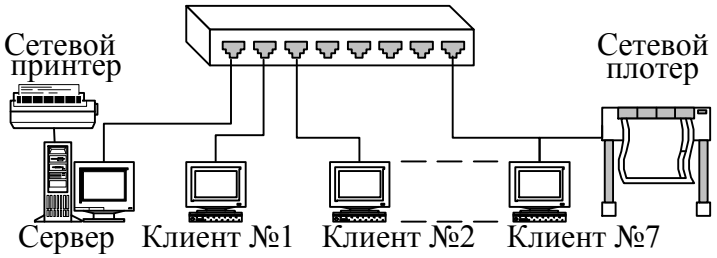


Рисунок 2 – Использование периферийного оборудования

С помощью сетей можно разделять ресурсы и информацию. Ниже перечислены основные задачи, которые решаются с помощью рабочей станции в сети, и которые трудно решить с помощью отдельного компьютера.

Компьютерная сеть позволит совместно использовать периферийные устройства, включая:

- принтеры;
- плоттеры;
- дисковые накопители;
- приводы CD-ROM;
- дисководы;
- стримеры;
- сканеры;
- факс-модемы;

Компьютерная сеть позволяет совместно использовать информационные ресурсы:

- каталоги;
- файлы;
- прикладные программы;
- игры;
- базы данных;
- текстовые процессоры.

Компьютерная сеть позволяет работать с многопользовательскими программами, обеспечивающими одновременный доступ всех пользователей к общим базам данных с блокировкой файлов и записей, обеспечивающей целостность данных. Любые программы, разработанные для стандартных ЛВС, можно использовать в других сетях.

Совместное использование ресурсов обеспечит существенную экономию средств и времени. Например, можно коллективно использовать один лазерный принтер вместо покупки принтера каждому сотруднику или беготни с дискетами к единственному принтеру при отсутствии сети.

Организация электронной почты. Можно использовать *ЛВС* как почтовую службу и рассылать служебные записки, доклады и сообщения другим пользователям.

Тема 2. Требования, предъявляемые к компьютерным сетям

Соответствие стандартам — это только одно из многих требований, предъявляемых к современным сетям. В этом разделе мы остановимся на некоторых других, не менее важных.

Самое общее пожелание, которое можно высказать в отношении работы сети — это выполнение сетью того набора услуг, для оказания которых она предназначена: например, предоставление доступа к файловым архивам или страницам публичных Web-сайтов Internet, обмен электронной почтой в пределах предприятия или в глобальных масштабах, интерактивный обмен голосовыми сообщениями IP-телефонии и т.п.

Все остальные требования — производительность, надежность, совместимость, управляемость, защищенность, расширяемость и масштабируемость — связаны с качеством выполнения этой основной задачи. И хотя все перечисленные выше требования весьма важны, часто понятие "качество обслуживания" (Quality of Service, QoS) компьютерной сети трактуется более узко: в него включаются только две самые важные характеристики сети — производительность и надежность.

Производительность

Потенциально высокая производительность — это одно из основных преимуществ распределенных систем, к которым относятся компьютерные сети. Это свойство обеспечивается принципиальной, но, к сожалению, не всегда практически реализуемой возможностью распределения работ между несколькими компьютерами сети.

Основные характеристики производительности сети:

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

Время реакции сети является интегральной характеристикой производительности сети с точки зрения пользователя. Именно эту характеристику имеет в виду пользователь, когда говорит: "Сегодня сеть работает медленно".

В общем случае время реакции определяется как интервал между возникновением запроса пользователя к какой-либо сетевой службе и получением ответа на него.

Очевидно, что значение этого показателя зависит от типа службы, к которой обращается пользователь, от того, какой пользователь и к какому серверу

обращается, а также от текущего состояния элементов сети — загруженности сегментов, коммутаторов и маршрутизаторов, через которые проходит запрос, загруженности сервера и т.п.

Поэтому имеет смысл использовать также и средневзвешенную оценку времени реакции сети, усредняя этот показатель по пользователям, серверам и времени дня (от которого в значительной степени зависит загрузка сети).

Время реакции сети обычно складывается из нескольких составляющих. В общем случае в него входит:

- время подготовки запросов на клиентском компьютере;
- время передачи запросов между клиентом и сервером через сегменты сети и промежуточное коммуникационное оборудование;
- время обработки запросов на сервере;
- время передачи ответов от сервера клиенту и время обработки получаемых от сервера ответов на клиентском компьютере.

Очевидно, что разложение времени реакции на составляющие пользователя не интересует — ему важен конечный результат. Однако для сетевого специалиста очень важно выделить из общего времени реакции составляющие, соответствующие этапам собственно сетевой обработки данных, — передачу данных от клиента к серверу через сегменты сети и коммуникационное оборудование.

Знание сетевых составляющих времени реакции позволяет оценить производительность отдельных элементов сети, выявить узкие места и при необходимости выполнить модернизацию сети для повышения ее общей производительности.

Производительность сети может характеризоваться также *скоростью передачи* трафика.

Скорость передачи трафика может быть мгновенной, максимальной и средней:

- средняя скорость вычисляется путем деления общего объема переданных данных на время их передачи, причем выбирается достаточно длительный промежуток времени — час, день или неделя;
- мгновенная скорость отличается от средней тем, что для усреднения выбирается очень маленький промежуток времени — например, 10 мс или 1 с;
- максимальная скорость — это наибольшая скорость, зафиксированная в течение периода наблюдения.

Чаще всего при проектировании, настройке и оптимизации сети используются такие показатели, как средняя и максимальная скорость. Средняя скорость, с которой обрабатывает трафик отдельный элемент или сеть в целом, позволяет оценить работу сети на протяжении длительного времени, в течение которого в силу закона больших чисел пики и спады интенсивности трафика компенсируют друг друга. Максимальная скорость позволяет оценить, как сеть будет справляться с

пиковыми нагрузками, характерными для особых периодов работы, например в утренние часы, когда сотрудники предприятия почти одновременно регистрируются в сети и обращаются к разделяемым файлам и базам данных. Обычно при определении скоростных характеристик некоторого сегмента или устройства в передаваемых данных не выделяется трафик какого-то определенного пользователя, приложения или компьютера — подсчитывается общий объем передаваемой информации. Тем не менее, для более точной оценки качества обслуживания такая детализация желательна, и в последнее время системы управления сетями все чаще позволяют ее выполнять.

Пропускная способность — максимально возможная скорость обработки трафика, определенная стандартом технологии, на которой построена сеть. Пропускная способность отражает максимально возможный объем данных, передаваемый сетью или ее частью в единицу времени.

Пропускная способность уже не является, подобно времени реакции или скорости прохождения данных по сети, пользовательской характеристикой, так как она говорит о скорости выполнения внутренних операций сети — передачи пакетов данных между узлами сети через различные коммуникационные устройства. Зато она непосредственно характеризует качество выполнения основной функции сети — транспортировки сообщений — и поэтому чаще используется при анализе производительности сети, чем время реакции или скорость.

Пропускная способностьизмеряется либо в битах в секунду, либо в пакетах в секунду.

Пропускная способность сети зависит как от характеристик физической среды передачи (медный кабель, оптическое волокно, витая пара) так и от принятого способа передачи данных (технология Ethernet, FastEthernet, ATM). Пропускная способность часто используется в качестве характеристики не столько сети, сколько собственно технологии, на которой построена сеть. Важность этой характеристики для сетевой технологии показывает, в частности, и то, что ее значение иногда становится частью названия, например, 10 Мбит/с Ethernet, 100 Мбит/с Ethernet.

В отличие от времени реакции или скорости передачи трафика пропускная способность не зависит от загруженности сети и имеет постоянное значение, определяемое используемыми в сети технологиями.

На разных участках гетерогенной сети, где используется несколько разных технологий, пропускная способность может быть различной. Для анализа и настройки сети очень полезно знать данные о пропускной способности отдельных ее элементов. Важно отметить, что из-за последовательного характера передачи данных различными элементами сети общая пропускная способность любого составного пути в сети будет равна минимальной из пропускных способностей составляющих элементов маршрута. Для повышения пропускной способности составного пути необходимо в первую очередь обратить внимание на самые медленные элементы. Иногда полезно оперировать общей пропускной

способностью сети, которая определяется как среднее количество информации, переданной между всеми узлами сети за единицу времени. Этот показатель характеризует качество сети в целом, не дифференцируя его по отдельным сегментам или устройствам.

Задержка передачи определяется как задержка между моментом поступления данных на вход какого-либо сетевого устройства или части сети и моментом появления их на выходе этого устройства.

Этот параметр производительности по смыслу близок ко времени реакции сети, но отличается тем, что всегда характеризует только сетевые этапы обработки данных, без задержек обработки конечными узлами сети.

Обычно качество сети характеризуют величинами максимальной задержки передачи и вариацией задержки. Не все типы трафика чувствительны к задержкам передачи, во всяком случае, к тем величинам задержек, которые характерны для компьютерных сетей, — обычно задержки не превышают сотен миллисекунд, реже — нескольких секунд. Такого порядка задержки пакетов, порождаемых файловой службой, службой электронной почты или службой печати, мало влияют на качество этих служб с точки зрения пользователя сети. С другой стороны, такие же задержки пакетов, переносящих голосовые или видеоданные, могут приводить к значительному снижению качества предоставляемой пользователю информации — возникновению эффекта "эха", невозможности разобрать некоторые слова, вибрации изображения и т. п.

Все указанные характеристики производительности сети достаточно независимы. В то время как пропускная способность сети является постоянной величиной, скорость передачи трафика может варьироваться в зависимости от загрузки сети, не превышая, конечно, предела, устанавливаемого пропускной способностью. Так в односегментной сети 10 Мбит/с Ethernet компьютеры могут обмениваться данными со скоростями 2 Мбит/с и 4 Мбит/с, но никогда — 12 Мбит/с.

Пропускная способность и задержки передачи также являются независимыми параметрами, так что сеть может обладать, например, высокой пропускной способностью, но вносить значительные задержки при передаче каждого пакета. Пример такой ситуации дает канал связи, образованный геостационарным спутником. Пропускная способность этого канала может быть весьма высокой, например 2 Мбит/с, в то время как задержка передачи всегда составляет не менее 0,24 с, что определяется скоростью распространения электрического сигнала (около 300000 км/с) и длиной канала (72000 км).

Надежность и безопасность

Одна из первоначальных целей создания распределенных систем, к которым относятся и вычислительные сети, состояла в достижении большей надежности по сравнению с отдельными вычислительными машинами.

Важно различать несколько аспектов надежности.

Для сравнительно простых технических устройств используются такие показатели надежности, как:

- среднее время наработки на отказ;
- вероятность отказа;
- интенсивность отказов.

Однако эти показатели пригодны для оценки надежности простых элементов и устройств, которые могут находиться только в двух состояниях — работоспособном или неработоспособном. Сложные системы, состоящие из многих элементов, кроме состояний работоспособности и неработоспособности, могут иметь и другие промежуточные состояния, которые эти характеристики не учитывают.

Для оценки надежности сложных систем применяется другой набор характеристик:

- готовность или коэффициент готовности;
- сохранность данных;
- согласованность (непротиворечивость) данных;
- вероятность доставки данных;
- безопасность;
- отказоустойчивость.

Готовность или коэффициент готовности (availability) означает период времени, в течение которого система может использоваться. Готовность может быть повышена путем введения избыточности в структуру системы: ключевые элементы системы должны существовать в нескольких экземплярах, чтобы при отказе одного из них функционирование системы обеспечивали другие.

Чтобы компьютерную систему можно было считать высоконадежной, она должна как минимум обладать высокой готовностью, но этого недостаточно. Необходимо обеспечить сохранность данных и защиту их от искажений. Кроме того, должна поддерживаться согласованность (непротиворечивость) данных, например если для повышения надежности на нескольких файловых серверах хранится несколько копий данных, то нужно постоянно обеспечивать их идентичность.

Так как сеть работает на основе механизма передачи пакетов между конечными узлами, одной из характеристик надежности является вероятность доставки пакета узлу назначения без искажений. Наряду с этой характеристикой могут использоваться и другие показатели: вероятность потери пакета (по любой из причин — из-за переполнения буфера маршрутизатора, несовпадения контрольной суммы, отсутствия работоспособного пути к узлу назначения и т. д.), вероятность искажения отдельного бита передаваемых данных, соотношение количества потерянных и доставленных пакетов.

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

распределенной системе это сделать гораздо сложнее, чем в централизованной. В сетях сообщения передаются по линиям связи, часто проходящим через общедоступные помещения, в которых могут быть установлены средства прослушивания линий. Другим уязвимым местом могут стать оставленные без присмотра персональные компьютеры. Кроме того, всегда имеется потенциальная угроза взлома защиты сети от неавторизованных пользователей, если сеть имеет выходы в глобальные общедоступные сети.

Еще одной характеристикой надежности является отказоустойчивость (fault tolerance). В сетях под отказоустойчивостью понимается способность системы скрыть от пользователя отказ отдельных ее элементов. Например, если копии таблицы базы данных хранятся одновременно на нескольких файловых серверах, пользователи могут просто не заметить отказа одного из них. В отказоустойчивой системе выход из строя одного из ее элементов приводит к некоторому снижению качества ее работы (деградации), а не к полному останову. Так, при отказе одного из файловых серверов в предыдущем примере увеличивается только время доступа к базе данных из-за уменьшения степени распараллеливания запросов, но в целом система будет продолжать выполнять свои функции.

Тема 3. Разделяемая среда передачи данных

Разделяемая среда – физическая среда передачи данных, к которой непосредственно подключено несколько передатчиков узлов сети. В каждый момент времени только один передатчик может использовать её для передачи данных приёмнику другого узла.

В сетях с разделяемой средой работа выполняется по следующему алгоритму:

1. Если в сети “тишина”, можно начать передачу пакета.
2. Если обнаружена коллизия, нужно прекратить передачу.
3. Через случайную паузу нужно повторить передачу испорченного пакета.

Существуют различные способы решения задачи организации совместного доступа к разделяемым линиям связи:

- централизованный подход управления доступом к разделяемой среде (доступом к каналу управляет специальное устройство — арбитр);
- децентрализованный подход управления доступом к разделяемой среде (не требуется наличие арбитра в сети).

Если обратиться к организации работы компьютера, то можно увидеть, что доступ к системной шине компьютера, которую совместно используют внутренние блоки компьютера, управляется централизованно — либо процессором, либо специальным арбитром шины.

Недостатки:

- система с разделяемой средой при увеличении количества подключенных к ней компьютеров будет работать все медленнее, поскольку пропускная способность линии делится между всеми компьютерами.
- ни один из компьютеров не может постоянно использовать линию. В каждый момент времени только один компьютер имеет право передавать данные в линию, так как в случае одновременной передачи несколькими компьютерами сигналы будут смешиваться и искажаться.

Неразделяемая среда (соединение точка-точка):

- полудуплексный режим работы – передача ведется в обоих направлениях, но с разделением по времени.
- полнодуплексный режим – передача может производиться одновременно с первым.

Коммутация - это соединение конечных узлов через сеть транзитных. *Маршрут* - это последовательность узлов, которые пройдут данные на пути от отправителя к получателю.

Обобщенная задача коммутации в общем виде может быть разбита на следующие взаимосвязанные составляющие:

- определение информационных потоков, для которых требуется прокладывать маршруты;
- маршрутизация потоков.
- продвижение потоков, то есть их распознавание и локальная коммутация на каждом транзитном узле.
- мультиплексирование и демультиплексирование потоков.

Определение информационных потоков

Информационным потоком (потоком данных) называется непрерывная последовательность данных, объединенных набором общих признаков, позволяющих выделить эти данных из общего сетевого трафика.

Логично предположить, что через один транзитный узел может проходить несколько маршрутов. Поэтому, транзитный узел должен уметь распознавать потоки данных для правильной их передачи на соответствующий сетевой интерфейс.

В качестве критерия определения потока данных может выступать адрес источника, тогда все данных от одного и того же компьютера будут являться потоком. Этот поток можно разделить на дочерние потоки, классифицируя данные по адресу назначения. Дальнейшую детализацию потоков данных можно выполнять, например, по номеру порта (используемой сетевой службе).

В англоязычной литературе имеется два термина для обозначения потока данных: data stream и data flow. Их различие состоит в том, что один термин подразумевает неравномерный поток данных (например, загрузка веб-страницы), в то время как второй - равномерный (например, потоковое аудио). В контексте этого пособия по умолчанию будет подразумеваться именно неравномерный поток данных. Если имеется в виду равномерный поток, то это будет оговорено.

Признаки потока могут быть глобальными или локальными. В то время как адреса источника и назначения являются глобальными, адрес сетевого интерфейса транзитного узла, с которого идет поток данных - локальный признак. Таким образом, транзитный узел может быть настроен таким образом, чтобы передавать данные, скажем, с интерфейса 1 на интерфейс 3. Таким образом, можно отделать потоки данных от разных узлов и использовать для них разные маршруты.

Существует еще и так называемая метка потока - особый тип признака, некое число, которое несут все данные потока. Метка потока так же может быть глобальной (не меняться при прохождении транзитных узлов) и локальной (подвергаться изменениям при переходе от одного транзитного узла к другому).

Задача *маршрутизации*, в свою очередь, делится на две составляющие задачи:

- определение маршрута.
- оповещение сети о выбранном маршруте.

Определение маршрута заключается в выборе последовательности транзитных узлов и их интерфейсов по которым будут передаваться данные из определенного источника в определенное назначение. Это довольно сложная задача, поскольку маршрутов может быть несколько, а выбрать надо один (наиболее оптимальный). Но даже если существует всего один маршрут в сети со сложной топологией, найти его тоже непросто. Критерии оптимальности часто строятся на количестве транзитных узлов, пропускной способности, надежности каналов и их загруженности. Маршрут может определяться администратором сети вручную на основе каких-либо его предпочтений и соображений, но в больших сетях со сложной топологией такой способ мало пригоден. В таком случае используются автоматические методы определения маршрутов. Для автоматизации процесса, компьютеры оснащаются дополнительным программным обеспечением, которое собирает информацию о сети и анализирует её. Затем, на основе полученных данных строится оптимальный маршрут.

Маршрут может строиться либо исходя из топологии сети (минимальное количество транзитных узлов), либо (если надо учитывать пропускную способность) - на основе так называемой метрики. Метрика представляет собой число, позволяющее судить о пропускной способности отдельных участков сети. Для задания метрики можно взять пропускную способность отдельного участка сети (например 100 Мб/с) и принять её равной 1. Таким образом, метрика других участков сети будет обратно пропорциональна пропускной способности с учетом выбранной точки отсчета. Например, метрика другого участка пропускной способностью 10 Мб/с будет равной 10. Таким образом, пройдя три транзитных узла мы получим метрику 3 против 10 в случае наличии прямого, но медленного маршрута.

Описанные методы определения маршрута упускают информацию о состоянии маршрутов. Таким образом, выбрав путь через три быстрых но загруженных транзитных узла можно проиграть, проигнорировав относительно медленный, но свободный маршрут.

Выбрав определенный маршрут, необходимо оповестить о нем все устройства сети. Делается это как правило посредством сообщений типа «данные потока n нужно передать на интерфейс X (или узлу 3)». Каждое устройство сети анализирует подобные сообщения и делает запись в специальной таблице коммутации. Этой таблицей в последствии будет руководствоваться узел при выборе интерфейса для передачи потока данных с определенными признаками.

Передача информации транзитным устройствам может осуществляться вручную (администратором) или автоматически по аналогии с определением маршрута.

Продвижение данных

Допустим, был определен маршрут и оповещена о нем сеть. Теперь настало время поговорить о продвижении данных. Для каждой пары абонентов эта операция может быть представлена как совокупность нескольких локальных операций коммутации на каждом из транзитных узлов. Отправитель передает данные на интерфейс, соответствующий маршруту, а транзитные узды передают данные с одного своего интерфейса на другой (выполняют коммутацию интерфейсов). Устройство, которое выполняет коммутацию интерфейсов называется коммутатором. Для того чтобы выполнить коммутацию, коммутатор должен распознать поток.

Важно отметить, что термины «коммутация» и «коммутатор» в телекоммуникационных сетях могут трактоваться неоднозначно. В общем смысле, коммутатор это устройство, способное передавать данные с одного сетевого интерфейса на другой. Некоторые способы коммутации и соответствующие устройства получили специальные названия. Например, в технологиях сетевого уровня IP и IPX, процесс коммутации называется «маршрутизация», а соответствующее устройство - «маршрутизатор». В то же время, для локальных сетей используются термины «коммутатор» и «коммутация». Для телефонных сетей термин «коммутатор» является синонимом телефонной станции.

Коммутатором может быть как аппаратное устройство, так и компьютер, использующий специализированное программное обеспечение. Более того, компьютер, используемый для коммутации может выполнять функции конечного или начального узла. Хорошей практикой является использование узлов, специально выделенных для коммутации, которые соединяются в так называемую коммутационную сеть. Остальные устройства подключаются уже к коммутационной сети.

Мультиплексирование и демультиплексирование

Как говорилось ранее, для выполнения коммутации, коммутатор должен определить поток данных. Определение потока должно выполняться независимо от того является поток «чистым» или «смешанным». В случае, если потом смешан с другими потоками, выполняется операция демультиплексирования.

Демультиплексирование - операция разделение одного смешанного потока на несколько чистых. Существует так же и обратная задача, которая называется мультиплексированием.

Мультиплексирование - операция объединения нескольких чистых потоков в один смешанный. Операция мультиплексирования выполняется тогда, когда коммутатору надо отправить несколько потоков данных на один сетевой интерфейс. Логично предположить, что технология мультиплексирования должна позволять последующее демультиплексирование. Самыми распространенными способами мультиплексирования являются разделение времени (когда для каждого потока данных отводится определенный промежуток времени использования канала) и частотное разделение канала, когда каждый поток данных использует свой

частотный диапазон. Функции мультиплексирования и демультиплексирования могут выполняться на каждом из сетевых интерфейсов коммутатора. Частным случаем коммутатора у которого все входные потоки данных коммутируются на один выходной интерфейс называется мультиплексор. Обратный случай называется демультиплексор.

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

На рис. 3 показан фрагмент сети, состоящий из трех коммутаторов. Коммутатор 1 имеет четыре сетевых интерфейса. На интерфейс 1 поступают данные с двух интерфейсов —3и4. Их надо передать в общий физический канал, то есть выполнить операцию мультиплексирования.

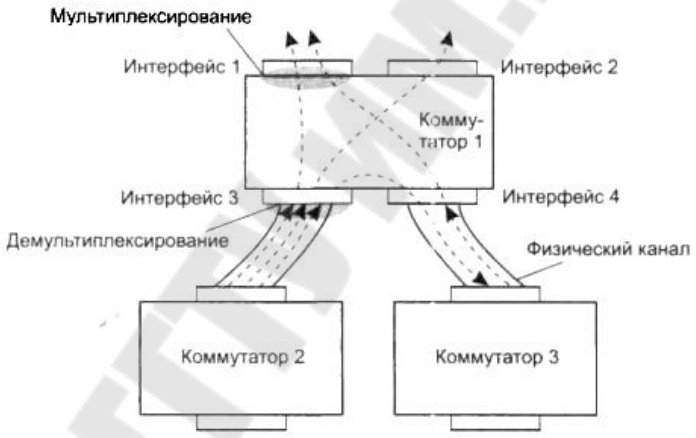


Рисунок 3 – Операция мультиплексирования/демультиплексирования

Одним из основных способов мультиплексирования потоков является разделение времени. При этом способе каждый поток время от времени (с фиксированным или случайным периодом) получает физический канал в полное свое распоряжение и передает по нему свои данные. Распространено также частотное разделение канала, когда каждый поток передает данные в выделенном ему частотное диапазоне.

Технология мультиплексирования должна позволять получателю такого суммарного потока выполнять обратную операцию — разделение (демультиплексирование) данных на слагаемые потоки. На интерфейсе 3 коммутатор выполняет демультиплексирование потока на три составляющих его подпотока. Один из них он передает на интерфейс 1, другой — на интерфейс 2, третий — на интерфейс 4. Вообще говоря, на каждом интерфейсе могут одновременно выполняться обе функции — мультиплексирование и демультиплексирование.

Тема 4. Распределённая обработка и распределённые системы

Современное производство требует высоких скоростей обработки информации, удобных форм ее хранения и передачи. Необходимо также иметь динамичные способы обращения к информации, способы поиска данных в заданные временные интервалы; реализовывать сложную математическую и логическую обработку данных. Управление крупными предприятиями, управление экономикой на уровне страны требуют участия в этом процессе достаточно крупных коллективов. Такие коллективы могут располагаться в различных районах города, в различных регионах страны и даже в различных странах. Для решения задач управления, обеспечивающих реализацию экономической стратегии, становятся важными и актуальными скорость и удобство обмена информацией, а также возможность тесного взаимодействия всех участвующих в процессе выработки управленческих решений.

В эпоху централизованного использования ЭВМ с пакетной обработкой информации пользователи вычислительной техники предпочитали приобретать компьютеры, на которых можно было бы решать почти все классы их задач. Однако сложность решаемых задач обратно пропорциональна их количеству, и это приводило к неэффективному использованию вычислительной мощности ЭВМ при значительных материальных затратах. Нельзя не учитывать и тот факт, что доступ к ресурсам компьютеров был затруднен из-за существующей политики централизации вычислительных средств в одном месте.

Принцип централизованной обработки данных (рис. 4) не отвечал высоким требованиям к надежности процесса обработки, затруднял развитие систем и не мог обеспечить необходимые временные параметры при диалоговой обработке данных в многопользовательском режиме. Кратковременный выход из строя центральной ЭВМ приводил к роковым последствиям для системы в целом, так как приходилось дублировать функции центральной ЭВМ, значительно увеличивая затраты на создание и эксплуатацию систем обработки данных.

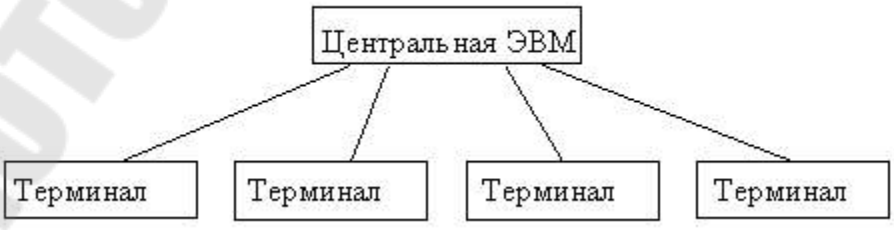


Рисунок 4 – Система централизованной обработки данных

Появление малых ЭВМ, микроЭВМ и, наконец, персональных компьютеров потребовало нового подхода к организации систем обработки данных, к созданию новых информационных технологий. Возникло логически обоснованное требование

перехода от использования отдельных ЭВМ в системах централизованной обработки данных к распределенной обработке данных (рис. 5).

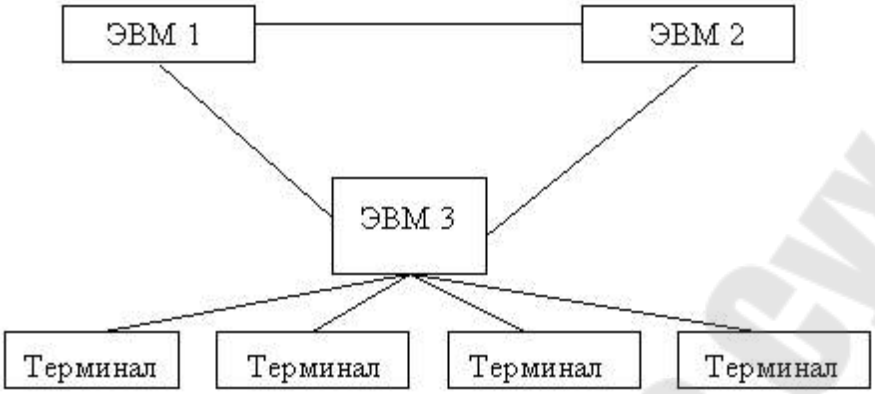


Рисунок 5 – Система распределенной обработки данных

Распределенная обработка данных - обработка данных, выполняемая на независимых, но связанных между собой компьютерах, представляющих распределенную систему.

Преимущества распределённой обработки данных:

- большое число взаимодействующих между собой пользователей, выполняющих функции сбора, регистрации, хранения, передачи и выдачи информации;

- снятие пиковых нагрузок с централизованной базы путем распределения обработки и хранения локальных баз данных на разных ЭВМ для обеспечения доступа информационного работника к вычислительным ресурсам сети ЭВМ;

- обеспечение симметричного обмена данными между удалёнными пользователями.

Для реализации распределенной обработки данных были созданы многомашинные ассоциации, структура которых разрабатывается по одному из следующих направлений:

- многомашинные вычислительные комплексы (МВК);
- компьютерные (вычислительные) сети.

Многомашинный вычислительный комплекс - группа установленных рядом вычислительных машин, объединенных с помощью специальных средств сопряжения и выполняющих совместно единый информационно-вычислительный процесс.

Многомашинные вычислительные комплексы могут быть:

- локальными при условии установки компьютеров в одном помещении, не требующих для взаимосвязи специального оборудования и каналов связи;
- дистанционными, если некоторые компьютеры комплекса установлены на значительном расстоянии от центральной ЭВМ и для передачи данных используются телефонные каналы связи.

Компьютерная (вычислительная) сеть - совокупность компьютеров и терминалов, соединенных с помощью каналов связи в единую систему, удовлетворяющую требованиям распределенной обработки данных.

Абоненты сети - объекты, генерирующие или потребляющие информацию в сети.

Абонентами сети могут быть отдельные ЭВМ, комплексы ЭВМ, терминалы, промышленные роботы, станки с числовым программным управлением и т.д. Любой абонент сети подключается к станции.

Станция - аппаратура, которая выполняет функции, связанные с передачей и приемом информации.

Совокупность абонента и станции принято называть абонентской системой. Для организации взаимодействия абонентов необходима физическая передающая среда.

Физическая передающая среда - линии связи или пространство, в котором распространяются электрические сигналы, и аппаратура передачи данных.

На базе физической передающей среды строится коммуникационная сеть, которая обеспечивает передачу информации между абонентскими системами.

Такой подход позволяет рассматривать любую компьютерную сеть как совокупность абонентских систем и коммуникационной сети.

РАЗДЕЛ 2. ЛОКАЛЬНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

Тема 5. Локальные вычислительные сети (ЛВС)

Локальная вычислительная сеть (ЛВС*) –совокупность аппаратного и программного* обеспечения, позволяющего объединить компьютеры в единую распределенную систему обработки и хранения информации. К аппаратному обеспечению можно отнести компьютеры с установленными на них сетевыми адаптерами, повторители, концентраторы, коммутаторы, мосты, маршрутизаторы, соединенные между собой сетевыми кабелями. К программному можно отнести сетевые операционные системы и протоколы передачи данных.

Задачи, решаемые ЛВС

1. Передача файлов. Электрический сигнал по кабелю из отдела в отдел движется быстрее, чем любой сотрудник с документом. Экономия бумаги и чернил принтера.

2. Разделение (совместное использование) файлов данных и программ. Отпадает необходимость дублировать данные на каждом компьютере.

3. Разделение (совместное использование) принтеров и другого оборудования. Значительно экономятся средства на приобретение и ремонт техники (сканеры, принтеры, модемы).

4. Электронная почта.

5. Координация совместной работы. При совместном решении задач каждый может оставаться на рабочем месте, но работать в «команде». Для менеджера проекта значительно упрощается задача контроля и координирования действий, т.к. сеть создает единое, легко наблюдаемое виртуальное пространство с большой скоростью взаимодействия территориально разнесенных участников.

6. Упорядочивание делопроизводства, контроль доступа к информации, защита информации. Чем меньше потенциальных возможностей потерять (забыть, положить не в ту папку) документ, тем меньше таких случаев будет. Гораздо легче найти документ на сервере (автоматический поиск, всегда известно авторство документа), чем в груде бумаг на столе. Сеть также позволяет проводить единую политику безопасности на предприятии, меньше полагаясь на сознательность сотрудников: всегда можно определить права доступа к документам и протоколировать все действия сотрудников.

С точки зрения организации взаимодействия персональных компьютеров локальные сети делят на одноранговые (Peet to Peet Network) и с выделенным сервером (Dedicated Server Network). Существуют также комбинированные сети, объединяющие свойства обоих типов сетей.

Компьютеры в *одноранговых* сетях могут выступать как в роли клиентов, так и в роли серверов. Так как все компьютеры в этом типе сетей равноправны, то

одноранговые сети не имеют централизованного управления[1] разделением ресурсов. Любой из компьютеров в этой сети может разделять свои ресурсы с любым компьютером из этой же сети. Одноранговые взаимоотношения также означают, что ни один компьютер не имеет ни высшего приоритета на доступ, ни повышенной ответственности за предоставление ресурсов в совместное использование[2].

Преимущества одноранговых сетей:

- они легки в установке и настройке;
- отдельные машины не зависят от выделенного сервера;
- пользователи в состоянии контролировать свои собственные ресурсы;
- недорогой тип сетей в приобретении и эксплуатации;
- не нужно никакого дополнительного оборудования или программного обеспечения, кроме операционной системы;
- нет необходимости нанимать администратора сети;
- хорошо подходит с количеством пользователей, не превышающих 10.

Недостатки одноранговых сетей:

- применение сетевой безопасности одновременно только к одному ресурсу;
- пользователи должны помнить столько паролей[3], сколько имеется разделенных ресурсов;
- необходимо производить резервное копирование отдельно на каждом компьютере, чтобы защитить все совместные данные;
- при получении доступа к ресурса, на компьютере, на котором этот ресурс расположен, ощущается падение производительности;
- не существует централизованной организационной схемы для поиска и управления доступом к данным.

Сети с выделенным сервером[4]

Компания Microsoft предпочитает термин Server-based. Сервер[5] представляет собой машину (компьютер), чьей основной задачей является реакция на клиентские[6] запросы. Серверы редко управляются кем-то непосредственно – только чтобы установить, настроить или обслуживать.

Достоинства сетей с выделенным сервером:

---

[1] **Centralized administration** (централизованное администрирование). Метод контроля над доступом к ресурсам сети и управления установкой и настройкой данных из одного места.

[2] **Workgroup model** (модель рабочих групп). Так Microsoft называет одноранговые сети, которые включают в себя один или более компьютеров под управлением Windows NT.

[3] Password (пароль). Секретная строка (должна быть трудноугадываемой), состоящая из букв, цифр и других символов, которая используется для идентификации конкретного пользователя и управления доступом к защищенным ресурсам.

[4] **Server-based network** (сеть с выделенным сервером). Тип или модель сети, в которой сетевой сервер предоставляет службы и ресурсы клиентским компьютерам и управляет доступом к этим службам и ресурсам.

[5] **Server** (сервер). Компьютер, который отвечает на запросы со стороны сетевых клиентов на доступ к службе или к ресурсу.

[6] **Client** (клиент). Сетевой компьютер, который запрашивает ресурсы или службы с другого компьютера, обычно сервера какого-нибудь типа.

– они обеспечивают централизованное управление учетными записями[7] пользователей, безопасностью и доступом, что упрощает сетевое администрирование;

– более мощное оборудование означает и более эффективный доступ к ресурсам сети;

– пользователям для входа в сеть нужно помнить только один пароль, что позволяет им получать доступ ко всем ресурсам, у которым имеет право;

– такие сети лучше масштабируются (растут) с ростом числа клиентов.

Недостатки сетей с выделенным сервером:

– неисправность сервера может сделать сеть неработоспособной, в лучшем случае – потеря сетевых ресурсов;

– такие сети требуют квалифицированного персонала для сопровождения сложного специализированного программного обеспечения;

– стоимость сети увеличивается, благодаря потребности в специализированном оборудовании и программном обеспечении.

Выбор архитектуры сети зависит от специфики организации, назначения сети и количества рабочих станций. От выбора типа сети зависит также и ее дальнейшее будущее: расширяемость, возможность использования того или иного ПО и оборудования, надежность сети и многое другое.

Топология компьютерных сетей.

При построении компьютерных сетей важным является выбор физической организации связей между отдельными компьютерами, т.е. топологии сети.

Топология сети - это ее геометрическая форма или схема физического расположения ПК по отношению друг к другу иих соединения каналами связи. Топология сети влияет на такие ее показатели, как надежность, расширяемость (наращиваемость), стоимость, задержку и пропускную способность.

При выборе топологии сети, наряду с чисто техническими проблемами передачи электрических сигналов, приходится решать и задачи экономного использования линий связи (1 км оптического волокна, например, стоит несколько тысяч долларов). Рассмотрим некоторые, наиболее часто встречающиеся топологии.

Полносвязная топология соответствует сети, в которой каждый компьютер связан со всеми остальными (Рис. 6а).

Полносвязная топология является громоздкой и малоэффективной, т.к. для каждой пары компьютеров выделяется отдельная электрическая линия связи и требуется большое количество коммутационных портов. Чаще всего этот вид топологии используется в глобальных сетях при небольших количествах компьютеров.

[7] **Account** (учетная запись). Информация о пользователе, которая может включать в себя имя владельца учетной записи, его пароль и принадлежащие пользователю права доступа к сетевым ресурсам.

Топология *общая шина* является достаточно распространенной топологией для локальных сетей (Рис. 6б). В этом случае компьютеры подключаются к одному общему кабелю (шине), по которому и происходит обмен информацией между компьютерами. Основными преимуществами общей шины являются дешевизна и простота разводки кабеля по отдельным помещениям. Серьезными недостатками такой топологии является низкая надежность, т.к любой дефект общего кабеля полностью парализует всю сеть, а так же невысокая производительность, поскольку в любой момент только один компьютер может передавать данные в сеть.

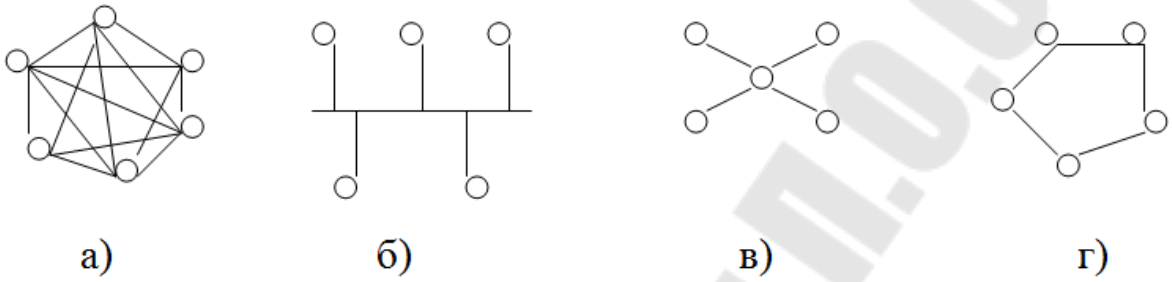


Рисунок 6 – Различные топологии компьютерных сетей

Топология *звезда* (Рис. 6в) предусматривает подключение каждого компьютера отдельным кабелем к общему устройству, называемому концентратором, который находится в центре сети. Концентратор служит для перенаправления передаваемой информации к одному или всем остальным компьютерам сети. По сравнению с общей шиной эта топология имеет более высокую надежность, т.к. неполадки с кабелем касаются лишь одного компьютера и только неисправность концентратора выводит из строя всю сеть. К недостатком топологии звезда можно отнести ее высокую стоимость ввиду необходимости установки дополнительного оборудования (концентратора). Кроме этого концентратор имеет ограниченное количество портов для подключения компьютеров. Поэтому для сетей с большим количеством компьютеров используется подключение нескольких концентраторов, иерархически соединенных между собой связями типа звезда. В настоящее время иерархическая звезда является самой распространенной топологией как в локальных, так и в глобальных компьютерных сетях.

В сетях с *кольцевой* топологией (Рис. 6г) данные передаются по кольцу от одного компьютера к другому, как правило, в одном направлении. Если компьютер распознает данные как «свои», то он их принимает. В сетях с кольцевой топологией всегда принимаются меры для обеспечения работоспособности сети при выходе из строя одного из компьютеров. Такие сети строятся всегда, если требуется контроль предаваемой информации, т.к. данные сделав полный оборот возвращаются к компьютеру- источнику.

Отметим, что по описанным типовым топологиям строятся, как правило, небольшие сети. Для крупных сетей характерно наличие произвольных связей между компьютерами, где можно, однако, выделить описанные выше топологии. Такие сети называются сетями со смешанной топологией.

На практике используются три базовых топологии - шина (EtherNet), кольцо (Token Ring), звезда (ArcNet).

Технические средства ЛВС - это ее сервер, рабочие станции, сетевые контроллеры, кабельные системы, соединители, разветвители, повторители, усилители, терминаторы-заглушки.

Сервер – это специально выделенный компьютер в сети, имеющий мощные ресурсы, высокую надежность, подключенный к источнику бесперебойного питания и оснащенный сетевой ОС. Он решает задачи управления сетью и поддержания ее работоспособности. Хранит общую информацию, обновляет ее копии у пользователей, проводит резервное копирование данных и т.д.

Рабочая станция - это любой, кроме сервера, компьютер, работающий в сети.

На практике используется два основных типа подключения компьютеров к ЛВС -тонкий Ethernet или витая пара. Тонкий Ethernet достаточно распространен как наиболее простой, дешевый, надежный вариант. Его элементами являются сетевые карты с "байонетной" розеткой, Т-коннекторы, N-коннекторы, терминаторы, кабель-коаксиал "тонкий Ethernet". Отрезки кабеля могут быть длиной 0,5-185 метров. Скорость передачи данных по такой сети - 10 (или 100) Мбит/с). Витая пара в реализации бывает нескольких типов. Различия между ними - в уровне помехозащищенности. Для сетей на витой паре требуется, кроме сетевых адаптеров еще и дополнительное устройство - концентратор (Hub). Эти сети имеют следующие преимущества - большая надежность, простое расширение сети, появление новых стандартов (в том числе на скорость передачи до 100 Мбит/с), не требует полной замены коммуникационных линий, большая производительность.

Дополнительно применяются также волоконно-оптические каналы связи и радиоканалы. Создание и быстрое совершенствование ПК типа Notebook привело к столь же быстрому развитию беспроводных технологий связи между ПК, в том числе в локальных сетях. Как правило, Notebook имеют стандартные порты или слоты. Благодаря этому можно легко и быстро присоединять Notebook к обычной ЛВС с помощью сетевых карт. Создание же беспроводного варианта ЛВС требует установки на каждую рабочую станцию или сервер специального устройства беспроводной связи. Сети реализуются либо в радиодиапазоне, либо в лазерных, либо в инфракрасных каналах связи.

Передача данных по сети регламентируется определенными правилами. Набор правил взаимодействия между компьютерами сети называют протоколами передачи данных, или сетевыми протоколами. Т.е. протокол- это "язык", на котором ПК разговаривают в сети между собой. Он устанавливается на них в виде программ-драйверов. Протоколы определяют формат, способ синхронизации, порядок

следования, методы обработки ошибок при передаче данных. Передача данных между компьютерами требует выполнения многих шагов.

Например, для передачи файла с одного компьютера на другой файл должен быть разбит на части, эти части должны быть определенным образом сгруппированы. Таким образом, компьютер, принимающий файл, должен получить дополнительную информацию о том, каким образом связаны между собой образованные группы, а также информацию о способе синхронизации, информацию, позволяющую корректировать ошибки, связанные с передачей данных, и т. д. Учитывая сложность осуществления коммуникаций между компьютерами, этот процесс обычно разбивается на шаги. Каждый такой шаг выполняется в соответствии со своими правилами, т. е. в соответствии со своим протоколом.

Тема 6. Принципы функционирования ЛВС

Главные функции сетей — передача информации и информационный доступ удаленных пользователей. Реализация этих функций возможна при условии использования адресатами открытых систем. Под открытой системой понимается система, архитектура которой общедоступна, т. е. существует набор аппаратных и программных средств, позволяющих коммутировать ее с другими платформами.

Для обеспечения согласованного функционирования различных компьютеров в сети, т. е. для связи между открытыми системами, было признано целесообразным разделить все требуемые для этого функции на группы соответственно решаемым задачам. Так, одна из групп обеспечивает передачу данных, другая — установку соединений, третья — выполнение программ конечных пользователей и т. д. Каждая такая функция (или набор функций), отвечающая за строго определенные действия, оформляется в виде протокола — набора правил, по которым производится обмен данными между компьютерами, независимо от их архитектуры и используемой ОС. Таким образом, протокол определяет следующие действия:

– синхронизация — механизм распознавания начала блока данных и его конца;

– инициализация — установление соединения между взаимодействующими партнерами по сеансу связи;

– блокирование — разбиение передаваемой информации на блоки данных строго определенной длины (включая опознавательные знаки начала блока и его конца);

– адресация — обеспечивает идентификацию пользователей, обменивающихся друг с другом информацией во время взаимодействия;

– обнаружение ошибок — использование различных методов, например установка битов четности для вычисления контрольных битов с целью проверки правильности передачи данных;

– нумерация блоков — текущая нумерация блоков, позволяющая установить ошибочно передаваемую или потерянную информацию;

– управление потоком данных — служит для распределения и синхронизации информационных потоков, например если не хватает места в буфере устройства данных или данные не достаточно быстро обрабатываются периферийными устройствами (принтеры, плоттеры и т. п.), сообщения и / или запросы накапливаются с целью дальнейшей обработки;

– методы восстановления — используют в случаях прерывания процесса передачи данных (ошибки, сбои оборудования, помехи в СПД и т. д.) для возврата в точку прерывания и повторной передачи информации;

– разрешение доступа — распределение, контроль и управление ограничениями доступа к данным (например, «только передача»).

Компьютеры для обмена данными через сеть должны использовать один и тот же протокол. Для координации совместной работы различных протоколов была разработана многоуровневая структура, где каждый протокол занимает свое место и взаимодействует с другими протоколами определенным образом. Можно сказать, что получился протокол взаимодействия протоколов, или система (модель) протоколов. Так, распределение протоколов по уровням для ЛВС определяется семиуровневой сетевой моделью взаимодействия открытых систем — OSI (Open Systems Interconnection), являющейся международным стандартом.

Международная Организация по Стандартам (International Standards Organization, ISO) разработала модель, которая четко определяет различные уровни взаимодействия систем, дает им стандартные имена и указывает, какую работу должен делать каждый уровень. Эта модель называется моделью взаимодействия открытых систем (Open System Interconnection, OSI) или моделью ISO/OSI.

В модели OSI взаимодействие делится на семь уровней или слоев (рис.7). Каждый уровень имеет дело с одним определенным аспектом взаимодействия. Таким образом, проблема взаимодействия декомпозирована на 7 частных проблем, каждая из которых может быть решена независимо от других. Каждый уровень поддерживает интерфейсы с выше- и нижележащими уровнями.

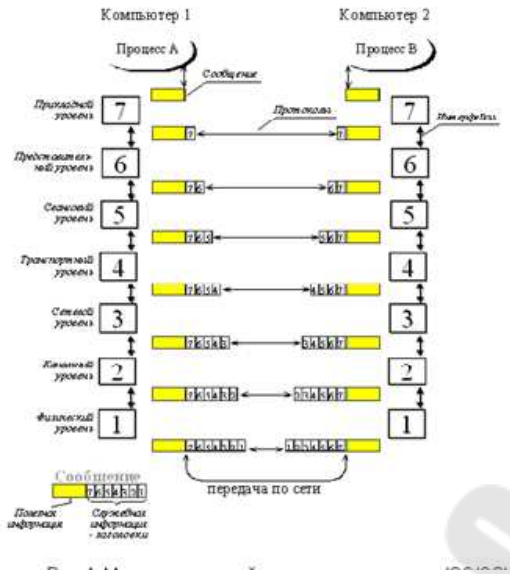


Рисунок 8 – Модель ISO/OSI

Модель OSI описывает только системные средства взаимодействия, не касаясь приложений конечных пользователей. Приложения реализуют свои собственные протоколы взаимодействия, обращаясь к системным средствам. Следует иметь в виду, что приложение может взять на себя функции некоторых верхних уровней модели OSI, в таком случае, при необходимости межсетевого обмена оно обращается напрямую к системным средствам, выполняющим функции оставшихся нижних уровней модели OSI.

Приложение конечного пользователя может использовать системные средства взаимодействия не только для организации диалога с другим приложением, выполняющимся на другой машине, но и просто для получения услуг того или иного сетевого сервиса.

Верхние уровни отвечают за взаимодействие прикладной программы с пользователем и поэтому ориентированы на прикладные процессы. Эти уровни не зависят от нижних и никак не связаны со способами доставки данных прикладным программам.

Нижние уровни обеспечивают передачу данных, включая упаковку, маршрутизацию, верификацию и формирование данных. Эти уровни игнорируют форматы данных.

Процесс передачи данных между прикладными программами осуществляется в три этапа: установление соединения, пересылка данных и разъединение.

Итак, пусть приложение обращается с запросом к прикладному уровню, например к файловому сервису. На основании этого запроса программное обеспечение прикладного уровня формирует сообщение стандартного формата, в которое помещает служебную информацию (заголовок) и, возможно, передаваемые данные. Затем это сообщение направляется представительному уровню.

Представительный уровень добавляет к сообщению свой заголовок и передает результат вниз сеансовому уровню, который в свою очередь добавляет свой заголовок и т.д.

Наконец, сообщение достигает самого низкого, физического уровня, который действительно передает его по линиям связи.

Когда сообщение по сети поступает на другую машину, оно последовательно перемещается вверх с уровня на уровень. Каждый уровень анализирует, обрабатывает и удаляет заголовок своего уровня, выполняет соответствующие данному уровню функции и передает сообщение вышележащему уровню.

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

Данные по сети передаются одинаковыми порциями, называемыми пакетами. В начале каждой порции находится некоторая служебная информация, называемая заголовком, а в конце добавляется завершитель. В локальной сети такая порция называется кадром или блоком данных (БД) и в общем виде может выглядеть следующим образом рисунок 8.

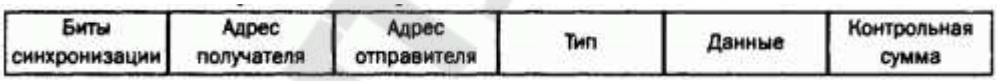


Рисунок 8 – Структура кадра

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

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

Канальный уровень. Одной из задач канального уровня является проверка доступности среды передачи. Другой задачей канального уровня является реализация механизмов обнаружения и коррекции ошибок. Для этого на канальном уровне биты группируются в наборы, называемые кадрами (frames). Канальный

уровень обеспечивает корректность передачи каждого кадра, помещая специальную последовательность бит в начало и конец каждого кадра, чтобы отметить его, а также вычисляет контрольную сумму, суммируя все байты кадра определенным способом и добавляя контрольную сумму к кадру. Когда кадр приходит, получатель снова вычисляет контрольную сумму полученных данных и сравнивает результат с контрольной суммой из кадра. Если они совпадают, кадр считается правильным и принимается. Если же контрольные суммы не совпадают, то фиксируется ошибка.

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

Сетевой уровень. Этот уровень служит для образования единой транспортной системы, объединяющей несколько сетей с различными принципами передачи информации между конечными узлами.

Сообщения сетевого уровня принято называть пакетами (packets). При организации доставки пакетов на сетевом уровне используется понятие "номер сети". В этом случае адрес получателя состоит из номера сети и номера компьютера в этой сети.

Для того, чтобы передать сообщение от отправителя, находящегося в одной сети, получателю, находящемуся в другой сети, нужно совершить некоторое количество транзитных передач (hops) между сетями, каждый раз выбирая подходящий маршрут. Таким образом, маршрут представляет собой последовательность маршрутизаторов, через которые проходит пакет.

Проблема выбора наилучшего пути называется маршрутизацией и ее решение является главной задачей сетевого уровня. Эта проблема осложняется тем, что самый короткий путь не всегда самый лучший. Часто критерием при выборе маршрута является время передачи данных по этому маршруту, оно зависит от пропускной способности каналов связи и интенсивности трафика, которая может изменяться с течением времени.

На сетевом уровне определяется два вида протоколов. Первый вид относится к определению правил передачи пакетов с данными конечных узлов от узла к маршрутизатору и между маршрутизаторами. Именно эти протоколы обычно имеют в виду, когда говорят о протоколах сетевого уровня. К сетевому уровню относят и другой вид протоколов, называемых протоколами обмена маршрутной информацией. С помощью этих протоколов маршрутизаторы собирают информацию о топологии межсетевых соединений. Протоколы сетевого уровня

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

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

Транспортный уровень. На пути от отправителя к получателю пакеты могут быть искажены или утеряны. Хотя некоторые приложения имеют собственные средства обработки ошибок, существуют и такие, которые предпочитают сразу иметь дело с надежным соединением. Работа транспортного уровня заключается в том, чтобы обеспечить приложениям или верхним уровням стека - прикладному и сеансовому - передачу данных с той степенью надежности, которая им требуется. Модель OSI определяет пять классов сервиса, предоставляемых транспортным уровнем.

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

Сеансовый уровень. Сеансовый уровень обеспечивает управление диалогом для того, чтобы фиксировать, какая из сторон является активной в настоящий момент, а также предоставляет средства синхронизации. Последние позволяют вставлять контрольные точки в длинные передачи, чтобы в случае отказа можно было вернуться назад к последней контрольной точке, вместо того, чтобы начинать все с начала. На практике немногие приложения используют сеансовый уровень, и он редко реализуется.

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

Прикладной уровень. Прикладной уровень - это в действительности просто набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые Web-страницы, а также организуют свою совместную работу, например, с помощью

протокола электронной почты. Единица данных, которой оперирует прикладной уровень, обычно называется сообщением (message).

Существует очень большое разнообразие протоколов прикладного уровня. Приведем в качестве примеров хотя бы несколько наиболее распространенных реализаций файловых сервисов: NCP в операционной системе Novell NetWare, SMB в Microsoft Windows NT, NFS, FTP и TFTP, входящие в стек TCP/IP.

Тема 7. Технические средства организации ЛВС

На самом нижнем уровне сетевых коммуникаций находится носитель, по которому передаются данные. В отношении передачи данных термин media [8] (носитель, среда передачи данных) может включать в себя как кабельные, так и беспроводные технологии.

Типы кабелей

Существует несколько различных видов кабелей, используемых в современных сетях. Различные сетевые ситуации могут потребовать различных типов кабелей.

Кабель типа «витая пара» – twisted pair

Представляет собой сетевой носитель, используемый во многих сетевых топологиях, включая Ethernet, ARCNet, IBM Token Ring.

Витая пара бывает двух видов.

1. Неэкранированная витая пара.

Имеется пять категорий неэкранированной витой пары. Они нумеруются по порядку возрастания качества от CAT1 до CAT5. Кабели более высокой категории обычно содержат больше пар проводников, и эти проводники имеют больше витков на единицу длины.

CAT1 – телефонный кабель, не поддерживает цифровой передачи данных.

CAT2 – представляет собой редко используемый старый тип неэкранированной витой пары. Он поддерживает скорость передачи данных до 4 Мбит/с.

CAT3 – минимальный уровень неэкранированной витой пары, требуемый для сегодняшних цифровых сетей, имеет пропускную способность 10 Мбит/с.

CAT4 – промежуточная спецификация кабеля, поддерживающая скорость передачи данных до 16 Мбит/с.

CAT5 – наиболее эффективный тип неэкранированной витой пары, поддерживающий скорость передачи данных до 100 Мбит/с.

[8] **Network medium** (сетевой носитель). Кабель — либо металлический, либо оптоволоконный, который связывает компьютеры в сети. Этот термин также используется для описания частот, используемых в беспроводных сетевых коммуникациях.

Кабели неэкранированной витой пары соединяют сетевую карту каждого компьютера с сетевой панелью или с сетевым концентратором с помощью соединителя RJ-45 для каждой точки соединения.

Примером такой конфигурации является стандарт на сеть Ethernet 10Base-T, который характеризуется кабелем неэкранированная витая пара (от CAT3 до CAT5) и использованием соединителя RJ-45.

Недостатки:

– чувствительность к помехам со стороны внешних электромагнитных источников;
– взаимное наложение сигнала между смежными проводами;
– неэкранированная витая пара уязвима для перехвата сигнала;
– большое затухание сигнала по пути (ограничение до 100 м).

2. Экранированная витая пара.

Имеет схожую конструкцию, что и предыдущая, подчиняется тому же 100-метровому ограничению. Обычно содержит в середине четыре или более пары скрученных медных изолированных проводов, а также электрически заземленную плетеную медную сетку или алюминиевую фольгу, создавая экран от внешнего электромагнитного воздействия.

Недостатки:

– кабель менее гибок;
– требует электрического заземления.

Коаксиальный кабель

Этот тип кабеля состоит из центрального медного проводника, более толстого, чем провода в кабеле типа витая пара. Центральный проводник покрыт слоем пенистого пластикового изолирующего материала, который в свою очередь окружен вторым проводником, обычно плетеной медной сеткой или алюминиевой фольгой. Внешний проводник не используется для передачи данных, а выступает как заземление.

Коаксиальный кабель может передавать данные со скорость до 10 Мбит/с на максимальное расстояние от 185 м до 500 м.

Двумя основными типами коаксиального кабеля, используемого в локальных сетях, является «Толстый Ethernet» (Thicknet) и «Тонкий Ethernet» (Thinnet).

1. Thinnet.

Также известен как кабель RG-58, является наиболее используемым. Он наиболее гибок из всех типов коаксиальных кабелей, имеет толщину примерно 6 мм. Он может использоваться для соединения каждого компьютера с другими компьютерами в локальной сети с помощью Т–коннектора, British Naval Connector (BNC)-коннектора и 50-Омных заглушек (terminator терминаторов). Используется в основном для сетей типа 10Base-2 Ethernet.

Эта конфигурация поддерживает передачу данных со скорость до 10 Мбит/с на максимальное расстояние до 185 м между повторителями.

2. Thicnet.

Является более толстым и более дорогим коаксиальным кабелем. По конструкции он схож с предыдущим, но менее гибок. Используется как основа для сетей 10Base-5 Ethernet. Этот кабель имеет маркировку RG-8 или RG-11, приблизительно 12 мм в диаметре. Он используется в виде линейной шины. Для подключения к каждой сетевой плате используется специальный внешний трансивер AUI (Attachment unit interface) и «вампир» (ответвление), пронизывающее оболочку кабеля для получения доступа к проводу.

Имеет толстый центральный проводник, который обеспечивает надежную передачу данных на расстояние до 500 м на сегмент кабеля. Часто используется для создания соединительных магистралей. Скорость передачи данных до 10 Мбит/с.

Оптоволоконный кабель

Обеспечивают превосходную скорость передачи информации на большие расстояния. Они не восприимчивы к электромагнитному шуму и подслушиванию.

Он состоит из центрального стеклянного или пластикового проводника, окруженного другим слоем стеклянного или пластикового покрытия, и внешней защитной оболочки. Данные передаются по кабелю с помощью лазерного или светодиодного передатчика, который посылает однонаправленные световые импульсы через центральное стеклянное волокно. Стеклянное покрытие помогает поддерживать фокусировку света во внутреннем проводнике. На другом конце проводника сигнал принимается фотодиодным приемником, преобразующем световые сигналы в электрический сигнал.

Скорость передачи данных для оптоволоконного кабеля достигает от 100 Мбит/с до 2Гбит/с. Данные могут быть надежно переданы на расстояние до 2 км без повторителя.

Световые импульсы двигаются только в одном направлении, поэтому необходимо иметь два проводника: входящий и исходящий кабели.

Этот кабель сложен в установке, является самым дорогим типом кабеля.

При планировании сети или расширении существующей сети необходимо четко рассмотреть несколько вопросов, касающихся кабелей: стоимость, расстояние, скорость передачи данных, легкость установки, количество поддерживаемых узлов.

Сравнение типов кабелей по скорости передачи данных, стоимости кабелей, сложности установки, максимального расстояния передачи данных представлено в таблице 1.

Количество узлов на сегмент и узлов в сети при построении сетей с различным использованием кабелей представлено в таблице 2.

Таблица 1 – Сравнительная характеристика кабелей

Тип	Скорость, Мбит/с	Длина, м	Установка	Цена
10Base-T	10	100	Легкая	Самый дешевый
100Base-T	100	100	Легкая	Дороже
Экранированная витая пара	16-155	100	Средней сложности	Еще дороже
10Base-2	10	185	Средней сложности	Недорогой
10Base-5	10	500	Сложнее, чем пред.	Дороже большинства кабелей
Оптоволокно	100-2000	2000	Самая сложная	Самый дорогой

Таблица 2 – Количество узлов в зависимости от типа сети

Тип сети	Узлов на сегмент	Узлов на сеть
10Base-T	1	1024
10Base-F	3	1024
100Base-T	1	1024
10Base-2 (5 сегментов, только в 3-х могут быть сервера)	30	900 (1024)
10Base-5 (5 сегментов, только в 3-х могут быть сервера)	100	1024

Сетевые компоненты

Существует множество сетевых устройств, которые можно использовать для создания, сегментирования и усовершенствования сети.

Сетевые карты[9]

Сетевой адаптер (Network Interface Card, *NIC*) - это периферийное устройство компьютера, непосредственно взаимодействующее со средой передачи данных, которая прямо или через другое коммуникационное оборудование связывает его с другими компьютерами. Это устройство решает задачи надежного обмена двоичными данными, представленными соответствующими электромагнитными сигналами, по внешним линиям связи. Как и любой контроллер компьютера, сетевой адаптер работает под управлением драйвера операционной системы.

В большинстве современных стандартов для локальных сетей предполагается, что между сетевыми адаптерами взаимодействующих компьютеров устанавливается специальное коммуникационное устройство (концентратор, мост, коммутатор или маршрутизатор), которое берет на себя некоторые функции по управлению потоком данных.

Сетевой адаптер обычно выполняет следующие функции:

[9] **Network Interface Card, NIC** (сетевой адаптер, сетевая карта). Плата адаптера персонального компьютера, которая дает возможность подключить компьютер к какому-либо типу сетевого носителя. Это устройство преобразует цифровую информацию в электрический сигнал для исходящих сетевых сообщений и входящие сигналы в их цифровой эквивалент.

– *Оформление передаваемой информации в виде кадра определенного формата*. Кадр включает несколько служебных полей, среди которых имеется адрес компьютера назначения и контрольная сумма кадра.

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

– *Кодирование последовательности бит кадра последовательностью электрических сигналов при передаче данных и декодирование при их приеме.* Кодирование должно обеспечить передачу исходной информацию по линиям связи с определенной полосой пропускания и определенным уровнем помех таким образом, чтобы принимающая сторона смогла распознать с высокой степенью вероятности посланную информацию.

– *Преобразование информации из параллельной формы в последовательную и обратно.* Эта операция связана с тем, что в вычислительных сетях информация передается в последовательной форме, бит за битом, а не побайтно, как внутри компьютера.

– *Синхронизация битов, байтов и кадров.* Для устойчивого приема передаваемой информации необходимо поддержание постоянного синхронизма приемника и передатчика информации.

Сетевые адаптеры различаются по типу и разрядности используемой в компьютере внутренней шины данных - ISA, EISA, PCI, MCA.

Сетевые адаптеры различаются также по типу принятой в сети сетевой технологии - Ethernet, Token Ring, FDDI и т.п. Как правило, конкретная модель сетевого адаптера работает по определенной сетевой технологии (например, Ethernet).

В связи с тем, что для каждой технологии сейчас имеется возможность использования различных сред передачи, сетевой адаптер может поддерживать как одну, так и одновременно несколько сред. В случае, когда сетевой адаптер поддерживает только одну среду передачи данных, а необходимо использовать другую, применяются трансиверы и конверторы.

Трансивер (приемопередатчик, transmitter+receiver) - это часть сетевого адаптера, его оконечное устройство, выходящее на кабель. В вариантах Ethernet'а оказалось удобным выпускать сетевые адаптеры с портом AUI, к которому можно присоединить трансивер для требуемой среды.

Вместо подбора подходящего трансивера можно использовать *конвертор*, который может согласовать выход приемопередатчика, предназначенного для одной среды, с другой средой передачи данных (например, выход на витую пару преобразуется в выход на коаксиальный кабель).

Повторители и усилители

Как говорилось ранее, сигнал при перемещении по сети, ослабевает. Чтобы предотвратить это ослабление, можно использовать повторители и (или) усилители, которые усиливают сигнал, проходящий через них.

Повторители (repeater) используются в сетях с цифровым сигналом для борьбы с затуханием (ослаблением) сигнала. Когда репитер получает ослабленный сигнал, он очищает этот сигнал, усиливает и посылает следующему сегменту.

Усилители (amplifier), хоть и имеют схожее назначение, используются для увеличения дальности передачи в сетях, использующих аналоговый сигнал. Это называется широкополосной передачей. Носитель делится на несколько каналов, так что разные частоты могут передаваться параллельно.

Обычно сетевая архитектура определяет максимальное количество повторителей, которые могут быть установлены в отдельной сети. Причиной этого является феномен, известный как «задержка распространения». Период, требуемый каждому повторителю для очистки и усиления сигнала, умноженный на число повторителей, может приводить к заметным задержкам передачи данных по сети.

Концентраторы

Концентратор (HUB) представляет собой сетевое устройство, действующее на физическом уровне сетевой модели OSI, служащее в качестве центральной точки соединения и связующей линии в сетевой конфигурации «звезда».

Существует три основных типа концентраторов:

– пассивные (passive);
– активные (active);
– интеллектуальные (intelligent).

Пассивные концентраторы не требуют электроэнергии и действуют как физическая точка соединения, ничего не добавляя к проходящему сигналу).

Активные требуют энергию, которую используют для восстановления и усиления сигнала.

Интеллектуальные концентраторы могут предоставлять такие сервисы, как переключение пакетов (packet switching) и перенаправление трафика (traffic riuting).

Мосты

Мост (bridge) представляет собой устройство, используемое для соединения сетевых сегментов. Мосты можно рассматривать как усовершенствование повторителей, так как они уменьшают загрузку сети: мосты считывают адрес сетевой карты (MAC address) компьютера-получателя из каждого входящего пакета данных и просматривают специальные таблицы, чтобы определить, что делать с пакетом.

Мост функционирует на канальном уровне сетевой модели OSI.

Мост функционирует как повторитель, он получает данные из любого сегмента, но он более разборчив, чем повторитель. Если получатель находится в том же физическом сегменте, что и мост, то мост знает, что пакет больше не нужен. Если получатель находится в другом сегменте, мост знает, что пакет надо переслать.

Эта обработка позволяет уменьшить загрузку сети, поскольку сегмент не будет получать сообщений, которые к нему не относятся.

Мосты могут соединять сегменты, которые используют разные типы носителей (10BaseT, 10Base2), а также с разными схемами доступа к носителю (Ethernet, Token Ring).

Маршрутизаторы

Маршрутизатор (router) представляет собой сетевое коммуникационное устройство, работающее на сетевом уровне сетевой модели, и может связывать два и более сетевых сегментов (или подсетей).

Он функционирует подобно мосту, но для фильтрации трафика он использует не адрес сетевой карты компьютера, а информацию о сетевом адресе, передаваемую в относящейся к сетевому уровню части пакета.

После получения этой информации маршрутизатор использует таблицу маршрутизации, чтобы определить, куда направить пакет.

Существует два типа маршрутизирующих устройств: статические и динамические. Первые используют статическую таблицу маршрутизации, которую должен создавать и обновлять сетевой администратор. Вторые – создают и обновляют свои таблицы сами.

Маршрутизаторы могут уменьшить загрузку сети, увеличить пропускную способность, а также повысить надежность доставки данных.

Маршрутизатором может быть как специальное электронное устройство, так и специализированный компьютер, подключенный к нескольким сетевым сегментам с помощью нескольких сетевых карт.

Он может связывать несколько небольших подсетей, использующих различные протоколы, если используемые протоколы поддерживают маршрутизацию. Маршрутизируемые протоколы обладают способностью перенаправлять пакеты данных в другие сетевые сегменты (TCP/IP, IPX/SPX). Не маршрутизируемый протокол – NetBEUI. Он не может работать за пределами своей собственной подсети.

Шлюзы

Шлюз (gateway) представляет собой метод осуществления связи между двумя и более сетевыми сегментами. Позволяет взаимодействовать несходным системам в сети (Intel и Macintosh).

Другой функцией шлюзов является преобразование протоколов. Шлюз может получить протокол IPX/SPX, направленный клиенту, использующему протокол

TCP/IP, на удаленном сегменте. Шлюз преобразует исходный протокол в требуемый протокол получателя.

Шлюз функционирует на транспортном уровне сетевой модели.

Тема 8. Сети Ethernet. Расчёт корректности конфигурации локальной сети Ethernet и Fast Ethernet

Ethernet - это самый распространенный на сегодняшний день стандарт локальных сетей, реализуемый на канальном уровне модели OSI. Общее количество работающих по протоколу Ethernet сетей оценивается в 5 миллионов, а количество компьютеров с установленными адаптерами Ethernet – более чем в 50 миллионов. Ethernet – это сетевой стандарт, разработанный фирмой Xerox в 1975 году и принятый комитетом IEEE (Institute of Electrical and Electronics Engineers).

Указанный стандарт использует метод разделения среды – метод CSMA/ CD (carrier- sense – multiply- acces with collision detection)- метод коллективного доступа с опознаванием несущей и обнаружением коллизий. Этот метод используется исключительно в сетях с топологией “общая шина”. Все компьютеры в такой топологии имеют доступ к общей шине, все компьютеры имеют возможность немедленно получить данные, которые любой из компьютеров начал передавать на общую шину. Простота подключения предопределяет успех технологии Ethernet. Базовый стандарт Ethernet предписывает передачу двоичной информации для всех вариантов физической среды со скоростью 10 Мбит/с.

Принцип работы Ethernet следующий.

Чтобы получить возможность передавать кадр компьютер должен убедиться, канал связи (среда) свободен. Это достигается прослушиванием основной гармоники сигнала, которая также называется несущей частотой (carrier- sense, CS). Признаком незанятости канала является отсутствие на ней несущей частоты (5 – 10 МГц). Если среда свободна, то компьютер начинает передавать кадр. Если в это время другой компьютер пробует начать передачу, но обнаруживает, что канал занят, он вынужден ждать, пока первый компьютер не прекратить передачу кадра.

После окончания передачи кадра все компьютеры вынуждены выдержать технологическую паузу в 9,6 мкс. Такая пауза необходима для приведения сетевых адаптеров в исходное состояние. Механизм прослушивания среды не гарантирует от возникновения такой ситуации, когда два или более компьютеров одновременно решают, что среда свободна и начинают передачу своих кадров. В этом случае возникает *коллизия,* так как оба кадры сталкиваются на общем кабеле и происходит искажение информации. (Рис. 10). Для возникновения коллизии не обязательно, чтобы несколько компьютеров начали передачу абсолютно одновременно, такая ситуация маловероятно. Гораздо вероятней, что коллизия возникает из-за того, что один компьютер начинает передачу кадра раньше другого, но до второго

компьютера сигнал первого просто не успевает дойти, когда он решает начать передачу. Другими словами, коллизии- это следствия распределенного характера сети. Чтобы отработать коллизию все компьютеры одновременно наблюдают за сигналами на кабеле.

Если передаваемые и наблюдаемые сигналы отличаются, то фиксируется коллизия. Для увеличения вероятности скорейшего обнаружения коллизии всеми компьютерами сети тот компьютер, который обнаружил коллизию прерывает передачу своего кадра и усиливает коллизию передачей в сеть специальной последовательности (4 байта), называемой jam- последовательностью.

Прекративший передачу компьютер должен сделать паузу в течение короткого случайного интервала времени, а затем снова предпринять попытку захвата канала и передачи кадра. Случайная пауза выбирается следующим образом:

Пауза = L x (интервал отсрочки) (1)

Интервал отсрочки равен 512 bt - битовым интервалам. В технологии Ethernet битовым интервалом называется интервал времени между появлением двух последовательных бит данных на кабеле. Для скорости канала 10 Мбит/ с величина битового интервала равна 0,1 мкс.

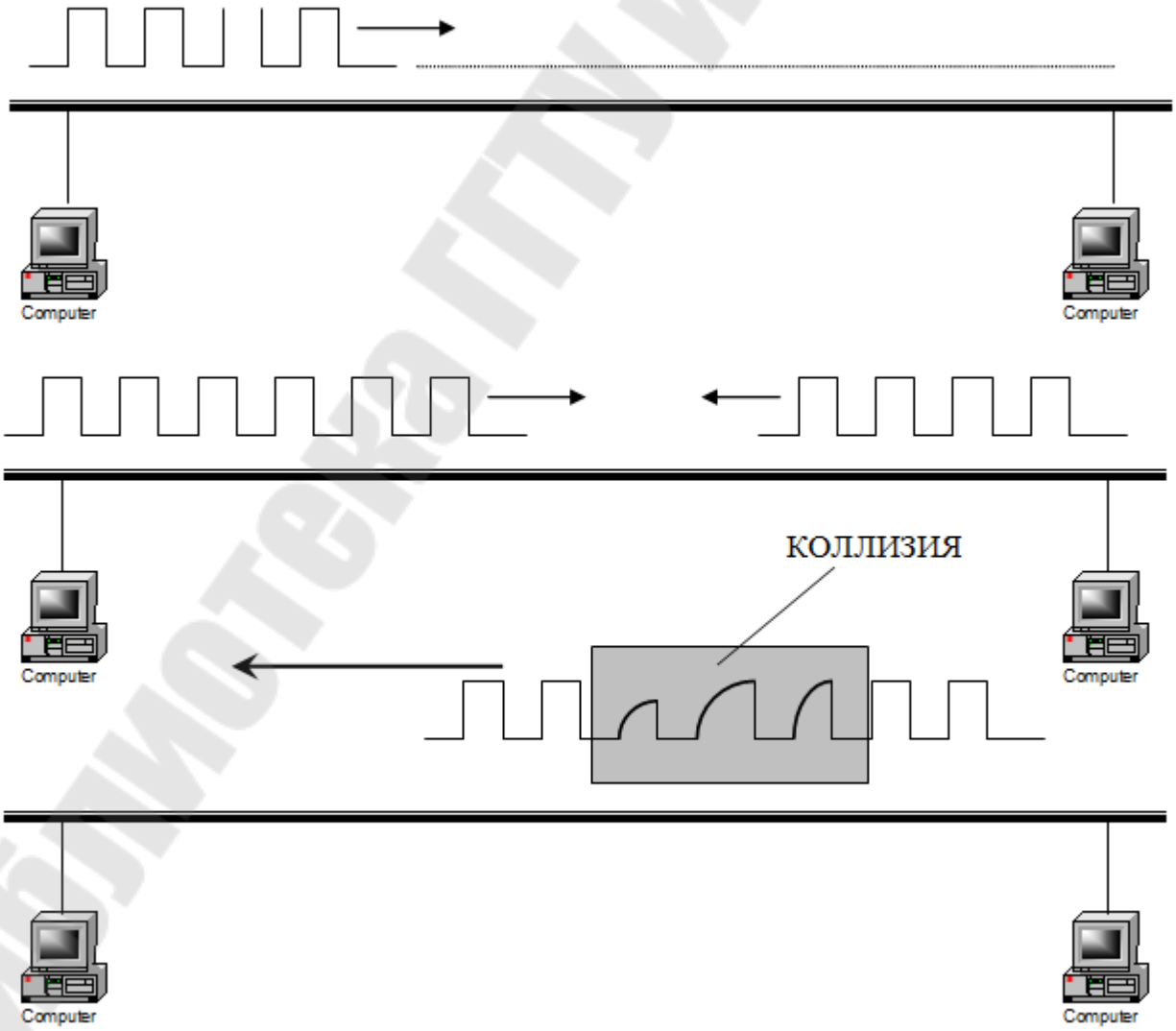


Рисунок 10 – Возникновение коллизии

L – представляет собой целое число, выбранное с равной вероятностью из диапазона 0, 2^N, где N- номер повторной попытки передачи данного кадра 1,2 ..10. После 10- й попытки интервал остается постоянным. Таким образом случайная пауза может быть от 0 до (1024 х 51,2 = 52428 мкс= 52,4 мс). Если 16 последовательных попыток передачи кадра вызывают коллизию, то передающий компьютер должен прекратить попытки и отбросить кадр. Из вышеописанного видно, что технология передачи Ethernet носит вероятностный характер и вероятность успешного получения в свое распоряжение общего канала зависит от загруженности сети, то есть от интенсивности возникновения компьютеров в передаче кадров.

Для надежного распознавания коллизий должно выполняться соотношение

$$T\,min \geq PDV \qquad (2)$$

где Tmin – время передачи кадра минимальной длины ,

PDV – время за которое сигнал коллизии успевает распространиться до самого дальнего компьютера сети. Так как в худшем случае сигнал должен пройти дважды между наиболее удаленными друг от друга компьютерами сети (в одну сторону проходит неискаженный сигнал, а на обратном пути распространяется уже искаженный сигнал), то это время называется временем двойного оборота (Path Delay Value). При выполнении этого условия передающий компьютер должен успевать обнаружить коллизию, которую вызвал переданный им кадр, еще как он закончить передачу этого кадра.

Выполнение условия (2) зависит, с одной стороны, от длины минимального кадра и пропускной способности сети, а с другой стороны, от длины кабельной системы сети и скорости распространения сигнала в кабеле (зависит от типа кабеля).

В стандарте Ethernet минимальная длина кадра вместе со служебной информацией установлена в размере 64 байта (46 байт данные + 18 байт служебная информация). Кроме этого в кадр входит 8 байт преамбулы для синхронизации адаптеров. Общая длина кадра составляет 8+64 = 72 байта или 72 х 8 = 576 бит. В стандартном 10 – мегабитном Ethernet время передачи кадра равно 576 х 0, 1 мкс = 57, 5 мкс. Межкадровый интервал устанавливается стандартом в размере 9, 6 мкс. В результате получаем, что период следования кадров минимальной длины составляет 57,5 + 9,6 = 67,1 мкс. (Рис. 11)

Отсюда максимальная пропускная способность стандарта Ethernet составляет 1/ 67,1 мкс= 14880 кадр/с.

Кадр максимальной длины в стандарте Ethernet составляет 1526 байт или 12208 бит. Максимальная пропускная способность Ethernet при работе с кадрами максимальной длины составляет 813 кадр/ с.

Под полезной пропускной способностью протокола понимается принимается скорость передачи полезной информации.

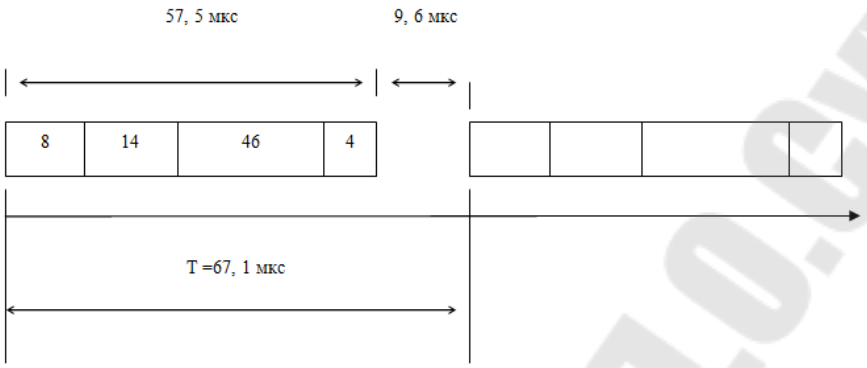


Рисунок 11 – Период следования кадров

Для кадров минимальной длины полезная пропускная способность равна

C= 14880 x 46 x 8 = 5, 48 Мбит/с (3)

что меньше 10 Мбит/ с.

Для кадров максимальной длины полезная пропускная способность:

C = 813 x 1526 x 8 = 9, 93 Мбит/с (4)

Таким образом передача кадров максимальной длины лучше всего обеспечивает стандарт Ethernet 10 Мбит/с.

Для соединения компьютеров между собой используются следующие стандартные физические линии связи:

- 10 Base– 5 – коаксиальный кабель диаметром 2,17 мм, называемый «толстым» коаксиалом
- 10 Base– 2 – коаксиальный кабель диаметром 0,89 мм называемый «тонким» коаксиалом
- 10Base –T – неэкранированная витая пара
- 10 Base – F – волоконно – оптический кабель

Расчет волновых сопротивлений кабелей и учет выражения (2) определяет в стандарте Ethernet максимальную длину сегмента (максимально возможное расстояние между компьютерами в сети), а также количество компьютеров в сегменте. *Сегментом* сети называется обособленная (физически или логически) группа компьютеров. В таблице 3 параметры сети Ethernet для основных используемых кабелей.

Таблица 1 - Параметры сети Ethernet для основных используемых кабелей.

	10 Base –5	10Base-2	10Base – T	10Base-F
Максимальная длина сегмента, м	500	185	100	2000
Максимальное число компьютеров в сегменте	100	30	1024	1024

В том случае, если сеть состоит из большого числа сегментов, они объединяются между собой с помощью специальных устройств, называемых *концентраторами.* Концентратор имеет несколько входов (портов), к которым подключаются компьютеры или другие концентраторы, и один выход.

Методика расчета конфигурации сети Ethernet

При конфигурировании сети Ethernet между конечными компьютерами разрешается использовать не более 4 концентраторов, 5 отрезков кабелей и 3- х нагруженных сегментов. *Нагруженным сегментом* называется концентратор с подключенными к нему компьютерами. *Не нагруженным* сегментом называется концентратор только с подключенными к нему другими концентраторами.

Это правило носит название *«правило 5-4-3».*

Важным показателем работоспособности сети является коэффициент загрузки сегмента сети S:

$$S = \frac{P \cdot m_i}{f} \qquad (5)$$

где P — количество компьютеров в сегменте сети

m_i — количество кадров в секунду, отправляемых в сеть i-м узлом;

f — максимально возможная пропускная способность сегмента, равная, как было указано выше 14880 кадр/с.

Имитационное моделирование сети Ethernet и исследование её работы с помощью анализаторов протоколов показали, что при коэффициенте загрузки S>0,5 начинается быстрый рост числа коллизий и, соответственно, увеличивается время ожидания доступа к сети.

Рекомендуемая величина коэффициента загрузки S для сети, использующих стандарт Ethernet, должна быть:

$S \leq 0,3$ (6)

Экспериментальные данные показали, что каждый из компьютеров передаёт в сеть в среднем от 500 до 1000 кадров в секунду. Таким образом, коэффициент загрузки сегмента равен:

$$S = \frac{P \cdot 1000}{14880} \qquad (7)$$

После расчета коэффициент загрузки сети Ethernet рассчитываются значения PDV, удовлетворяющего условию:

$$PDV \leq 575 \qquad (8)$$

а также сокращения межкадрового интервала PVV (Path Variability Value):

$$PVV \leq 49 \qquad (9)$$

Указанные значения являются экспериментальными и получены для различных физических сред стандарта Ethernet.

Общее значение PDV равно сумме всех значений PDVi на каждом участке, а значение PDVi равно сумме задержек, вносимой i- базой сегмента и задержкой, вносимой кабелем:

$PDV = \Sigma PDVi$, где $PDVi = ti\ базы + ti\ кабеля$ (10)

В свою очередь

$ti\ кабеля = L\ i\ x\ bt\ i$ (11)

Аналогичным образом сокращение межкадрового интервала равно:

$PVV = \Sigma PVVi$ (12)

причем в расчет не включается правый сегмент.

В таблице 4 приведены значения затуханий, для расчета PDV вносимые элементами сети в битовых интервалах bt. Интервалы bt приведены в таблице уже умноженные на 2, т.к. высчитывается двойное время оборота сигнала (по определению PDV)

Таблица 4 – значения затуханий, для расчета PDV

Тип сегмента	База левого сегмента, bt	База промежуточного сегмента, bt	База правого сегмента, bt	Задержка среды на 1 м
10Base-5	11,8	46,5	169,5	0,0866
10Base-2	11,8	46,5	169,5	0,1026
10bas e-T	15,3	42,0	165,0	0,113
10Base-F	12,3	33,5	156,5	0,1

В таблице 5 приведены значения затуханий, для расчета PDV

Таблица 5 – Значения затуханий, для расчета PDV

Тип сегмента	Левый сегмент, bt	Промежуточный сегмент,bt
10Base-5	16	11
10Base-2	16	11
10base-T	10,5	8
10Base-F	10,5	8

В таблицах используются понятия *левый сегмент, правый сегмент и промежуточный сегмент*.

Кроме затуханий, вносимых физическими линиями связи, подключенные к концентраторам , сегменты вносят собственные задержки, называемые *базами*.

Пример:

Расчитаем сеть, педставленную на рис. 3. Пеедающий компьютер находится в левом сегменте. Сигнал проходит через промежуточные сегменты и доходит до принимающего компьютера, который находится в правом сегменте. Количество компьютеров в каждом сегменте обеспечивает коэффициент загрузки S< 0,3.

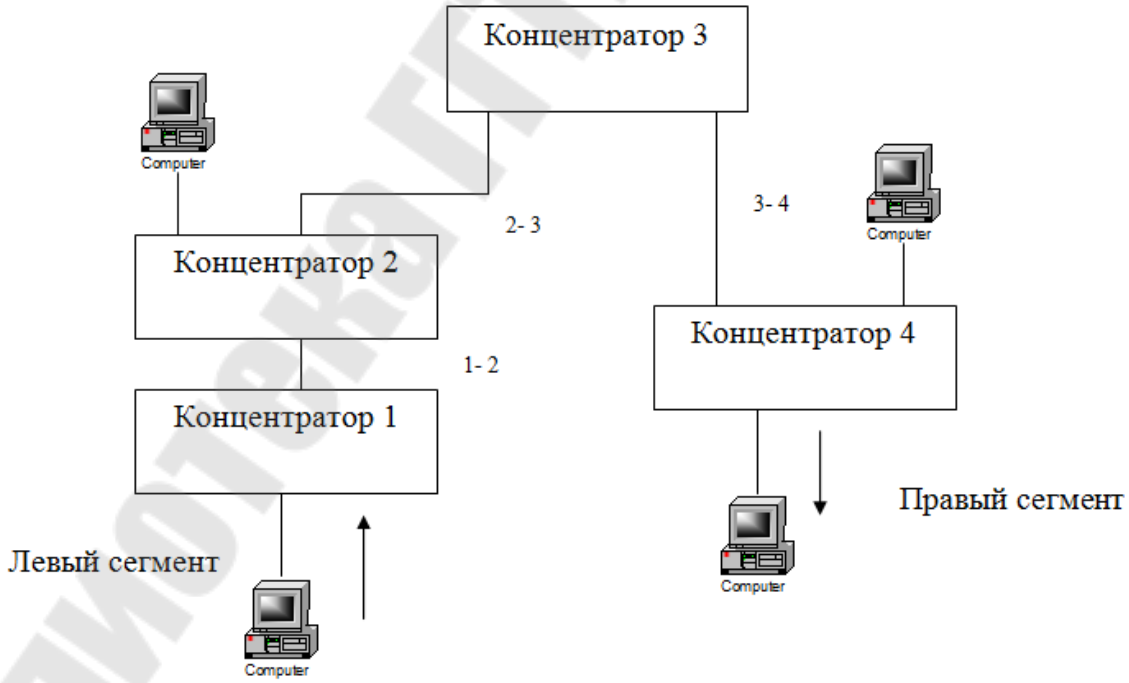


Рисунок 12 – Пример расчета конфигурации сети Ethernet

Пусть физические среды и расстояние между концентраторами следующие

Участок между концентраторами	Физическая среда	Длина, м
Левый сегмент	10Base –T	90
1-2	10 Base –2	130
2-3	10Base-F	1000
3-4	10 Base –5	200
Правый сегмент	10Base –T	100

Решение

1) Проверка выполнения «правила 4-5 -3»

Сеть содержит 4 концентратора, 5 отрезков кабелей и 3 нагруженных сегмента (концентраторы 1,2,4). «Правило 4-5-3» выполняется

а) Расчет PDV

- левый сегмент PDV1 = 15,3 + 90 x 0,113 = 25,47
- промежуточный сегмент 1-2 PDV2 = 46,5 + 130 x 0,1026 = 59,84
- промежуточный сегмент 2-3 PDV3 = 33,5 + 1000 x 0,1= 133,50
- промежуточный сегмент 3-4 PDV4 = 46,5 + 200 x 0,0866 = 63,82
- правый сегмент PDV5 = 165 + 100 x 0,113 = 176,30

Таким образом, PDV сети равно:

PDV = 25,47 + 59,84+ 133,50+ 63,82 + 176,30 = 458,93 < 575

Значение рассчитанного PDV меньше допустимой величины. Это значит, что сеть является работоспособной по критерию времени двойного оборота сигнала.

б*) расчет PVV*

Из таблицы 2 выбираем:

- левый сегмент PVV1 = 10,5
- промежуточный сегмент 1-2 PVV2 = 11
- промежуточный сегмент 2-3 PVV3 = 8
- промежуточный сегмент 3-4 PVV4 = 11

В результате получим значение:

PVV = 10,5 + 11 + 8 + 11 = 40,5 < 49

Значение рассчитанного PVV меньше допустимой величины. Это значит, что сеть является работоспособной также и по критерию сокращение межкадрового интервала.

Отметим, что в случае не выполнения условий (8) , (9) необходимо менять конфигурацию сети или уменьшать длины соединительных кабелей и их типы.

При использовании в сети вместо концентраторов специальных устройств *коммутаторов* общие PDV и PVV сети не суммируется по всем участкам (из- за

того, что коммутаторы физически разделяют сеть), а условия (8), (9) проверяется по каждому участку.

Тема 9. Беспроводные компьютерные сети

В зарубежной литературе принято обозначать беспроводную сеть, как Wireless Area Network. Для сети с небольшим радиусом действия, например в пределах одного помещения, используют обозначение (Wireless LAN). Это вид вычислительных сетей, который использует для связи и передачи данных между узлами и компонентами высокочастотные радиоволны.

Методы беспроводной передачи данных являются более удобной формой. Беспроводные технологии различаются по типам сигналов, частоте, расстоянию передачи.

Тремя главными типами беспроводной передачи данных являются: радиосвязь, связь в микроволновом диапазоне, инфракрасная связь.

Радиосвязь

Технологии радиосвязи пересылают данные на радиочастотах и практически не имеет ограничений на дальность. Используется для соединения локальных сетей на больших географических расстояниях.

Недостатки:

- радиопередача имеет высокую стоимость,
- подлежит государственному регулированию,
- крайне чувствительна к электронному или атмосферному влиянию,
- подвержена перехвату, поэтому требует шифрования.

Связь в микроволновом диапазоне

Поддерживает передачу данных в микроволновом диапазоне, использует высокие частоты и применяется как на коротких расстояниях, так и в глобальной коммуникациях.

Ограничение: передатчик и приемник должны быть в зоне прямой видимости друг друга.

Широко используется в глобальной передаче информации с помощью спутников и наземных спутниковых антенн.

Инфракрасная связь

Функционирует на высоких частотах, приближающихся к частотам видимого света. Могут быть использованы для установления двусторонней или широковещательной передачи данных на близкие расстояния. Обычно используют светодиоды для передачи инфракрасных волн приемнику.

Эти волны могут быть физически заблокированы и испытывают интерференцию с ярким светом, поэтому передача ограничена малыми расстояниями.

Беспроводные компьютерные сети — это технология, позволяющая создавать вычислительные сети, полностью соответствующие стандартам для обычных проводных сетей (например, Ethernet), без использования кабельной проводки. В качестве носителя информации в таких сетях выступают радиоволны СВЧ-диапазона.

Существуют два вида беспроводных сетей: ad-hoc и инфраструктурная сеть.

Сеть ad-hoc(читается эд-хок) это наиболее простая беспроводная сеть, которая создается посредством объединения двух или более беспроводных клиентов без наличия точки доступа. Все клиенты внутри сети ad-hoc равноправны и позволяет организовать обмен файлами и информацией между устройствами без затрат и сложностей, связанных с приобретением и настройкой точки доступа.

Инфраструктурная сеть– обладает точкой доступа, управляющей обменом данных в пределах беспроводной соты (зоны покрытия). Точка доступа определяет, какие узлы и в какое время могут устанавливать связь. Такой режим работы сети наиболее популярен. При такой форме организации беспроводных сетей отдельные беспроводные устройства не могут взаимодействовать между собой напрямую. Чтобы эти устройства могли взаимодействовать между собой, им необходимо разрешение от точки доступа. Точка доступа управляет всеми взаимодействиями и обеспечивает равный доступ к сети всем устройствам.

Как было упомянуто, точка доступа имеет ограниченную зону покрытия. Для увеличения зоны покрытия, можно установить несколько точек доступа с общим SSID. В таком случае, следует помнить, что для того, чтобы переход между сотами был возможен без потери сигнала, зоны покрытия соседних точек доступа должны пересекаться между собой примерно на 10%. Это позволяет клиенту подключаться ко второй точке доступа перед тем, как отключиться от первой точки доступа.

Достоинства и недостатки использования беспроводной сети

Достоинства:

- избавление от кабелей (самый большой плюс);
- минимум монтажных работ;
- могут обслуживаться места, где нельзя проложить кабель (например, в зданиях, имеющих историческую ценность);
- избавляет от привязки к конкретному месту;
- легкость переезда всего оборудования;
- позволяет иметь доступ к сети мобильным устройствам.

Недостатки:

- если сеть построена или пролегает через открытое пространство (улицы, дома, Ж/Д пути и пр.) возможны помехи как от других линий связи, так и от плохой погоды (дождь, снегопад), для устранения данных помех придется докупать дополнительное оборудование;
- при незащищенном использование возможен легкий доступ извне, в радиусе действия сетей Wi-Fi. Для предотвращения этого существует шифрование канала, которое нужно обязательно использовать при создании сети;
- стоимость, чаще всего, получается дороже, чем воздвигнуть проводную сеть.

Большим недостатком есть протоколы кодирования. Например, если вы пользуетесь беспроводным интернетом в общественном месте, вся ваша информация доступна третьей стороне. Все данные, включая даже те, что вы храните у себя на жестком диске ноутбука.

Из-за того, что количество пользователей беспроводного доступа с каждым днем становится больше, увеличивается и нагрузка на каналы, по которым передаются данные. Со временем, если на эту проблему не обратить должного внимания, одни пользователи будут мешать другим.

Кроме маршрутизаторов, перегружать беспроводные сети могут и другие устройства, такие как радиотелефоны, микроволновые печи (создают помехи при передаче данных), а также устройства Bluetooth. Чем больше город, тем больше возможность перегрузки сетей.

Беспроводные маршрутизаторы имеют свой диапазон работы, радиусом от 45 до 90 метров. Диапазон можно расширить, купив антенну.

Тема 10. Передача данных по сети.

Уровни модели OSI

Ниже перечислены (в направлении сверху вниз) уровни модели OSI и указаны их общие функции.

Уровень приложения (Application) - интерфейс с прикладными процессами.

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

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

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

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

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

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

Информация в локальных сетях, как правило, передается отдельными порциями, кусками, называемыми в различных источниках пакетами, кадрами или блоками. Использование пакетов связано с тем, что в сети, как правило, одновременно может происходить несколько сеансов связи, то есть в течение одного и того же интервала времени могут идти два или больше процессов передачи данных между различными парами абонентов. Пакеты как раз и позволяют разделить во времени сеть между передающими информацию абонентами.

Если бы вся требуемая информация передавалась сразу, непрерывно, без разделения на пакеты, то это привело бы к монопольному захвату сети одним из абонентов на довольно продолжительное время. Все остальные абоненты вынуждены были бы ждать окончания передачи всей информации, что в ряде случаев могло бы потребовать десятков секунд и даже минут (например, при копировании содержимого целого жесткого диска). Чтобы уравнять в правах всех абонентов, а также примерно уравнять время доступа к сети и интегральную скорость передачи информации для всех абонентов, как раз и используются пакеты (кадры). Длина пакета зависит от типа сети, но обычно она составляет от нескольких десятков байт до нескольких килобайт.

Структура пакета определяется прежде всего аппаратными особенностями данной сети, выбранной топологией и типом среды передачи информации, а также существенно зависит от используемого протокола (порядка обмена информацией). Строго говоря, в каждой сети структура пакета индивидуальна. Но существуют некоторые общие принципы формирования пакета, определяемые характерными особенностями обмена информацией по любым локальным сетям.

Модель OSI описывает только системные средства взаимодействия, реализуемые операционной системой, системными утилитами, системными аппаратными средствами. Модель не включает средства взаимодействия приложений конечных пользователей. Свои собственные протоколы взаимодействия приложения реализуют, обращаясь к системным средствам. Поэтому необходимо различать уровень взаимодействия приложений и прикладной уровень.

Следует также иметь в виду, что приложение может взять на себя функции некоторых верхних уровней модели OSI. Например, некоторые СУБД имеют встроенные средства удаленного доступа к файлам. В этом случае приложение, выполняя доступ к удаленным ресурсам, не использует системную файловую службу; оно обходит верхние уровни модели OSI и обращается напрямую к системным средствам, ответственным за транспортировку сообщений по сети, которые располагаются на нижних уровнях модели OSI.

Итак, пусть приложение обращается с запросом к прикладному уровню, например к файловой службе. На основании этого запроса программное обеспечение прикладного уровня формирует сообщение стандартного формата. Обычное сообщение состоит из заголовка и поля данных. Заголовок содержит служебную информацию, которую необходимо передать через сеть прикладному уровню машины-адресата, чтобы сообщить ему, какую работу надо выполнить. В нашем случае заголовок, очевидно, должен содержать информацию о месте нахождения файла и о типе операции, которую необходимо над ним выполнить. Поле данных сообщения может быть пустым или содержать какие-либо данные, например те, которые необходимо записать в удаленный файл. Но для того чтобы доставить эту информацию по назначению, предстоит решить еще много задач, ответственность за которые несут нижележащие уровни.

После формирования сообщения прикладной уровень направляет его вниз по стеку представительному уровню. Протокол представительного уровня на основании информации, полученной из заголовка прикладного уровня, выполняет требуемые действия и добавляет к сообщению собственную служебную информацию - заголовок представительного уровня, в котором содержатся указания для протокола представительного уровня машины-адресата. Полученное в результате сообщение передается вниз сеансовому уровню, который в свою очередь добавляет свой заголовок, и т. д. (Некоторые реализации протоколов помещают служебную информацию не только в начале сообщения в виде заголовка, но и в

конце, в виде так называемого «концевика».) Наконец, сообщение достигает нижнего, физического уровня, который собственно и передает его по линиям связи машине-адресату. К этому моменту сообщение «обрастает» заголовками всех уровней (рис. 13).

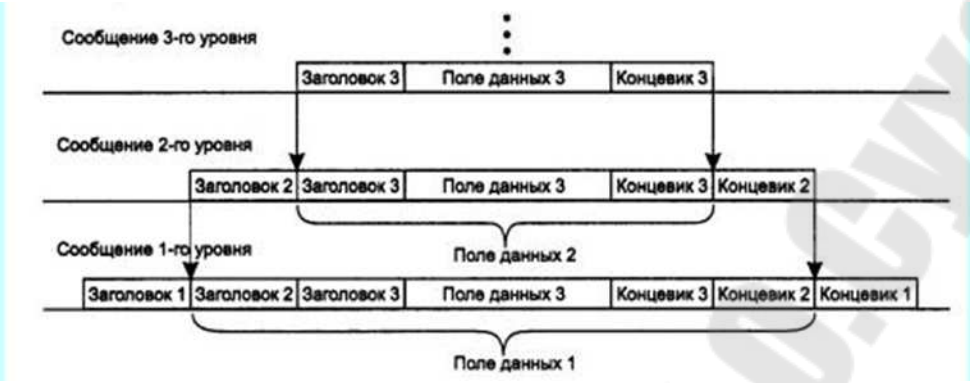


Рисунок 13 – Вложенность сообщений различных уровней

Когда сообщение по сети поступает на машину - адресат, оно принимается ее физическим уровнем и последовательно перемещается вверх с уровня на уровень. Каждый уровень анализирует и обрабатывает заголовок своего уровня, выполняя соответствующие данному уровню функции, а затем удаляет этот заголовок и передает сообщение вышележащему уровню.

Наряду с термином *сообщение (message)* существуют и другие термины, применяемые сетевыми специалистами для обозначения единиц данных в процедурах обмена. В стандартах ISO для обозначения единиц данных, с которыми имеют дело протоколы разных уровней, используется общее название *протокольный блок данных (Protocol Data Unit, PDU)*. Для обозначения блоков данных определенных уровней часто используются специальные названия: кадр (frame), пакет (packet), дейтаграмма (datagram), сегмент (segment).

В модели OSI различаются два основных типа протоколов. В протоколах с *установлением соединения (connection-oriented)* перед обменом данными отправитель и получатель должны сначала установить соединение и, возможно, выбрать некоторые параметры протокола, которые они будут использовать при обмене данными. После завершения диалога они должны разорвать это соединение. Телефон - это пример взаимодействия, основанного на установлении соединения.

Вторая группа протоколов - протоколы *без предварительного установления соединения (connectionless)*. Такие протоколы называются также *дейтаграммными* протоколами. Отправитель просто передает сообщение, когда оно готово. Опускание письма в почтовый ящик - это пример связи без предварительного установления соединения. При взаимодействии компьютеров используются протоколы обоих типов.

Инкапсуля́ция вкомпью́ терныхсетя́ х — это метод построения модульных сетевых протоколов, при котором логически независимые функции сети

абстрагируются от нижележащих механизмов путём включения или инкапсулирования этих механизмов в более высокоуровневые объекты. Например, когда приложению требуется послать сообщение с помощью UDP, то производится последовательность действий:

– в первую очередь приложение заполняет специальную структуру данных, в которой указывает информацию о получателе (сетевой протокол, IP-адрес, порт UDP);

– передаёт сообщение, его длину и структуру с информацией о получателе обработчику протокола UDP (транспортный уровень);

– обработчик UDP формирует датаграмму, в которой в качестве данных выступает сообщение, а в заголовках находится UDP-порт получателя (а также другие данные);

– обработчик UDP передаёт сформированную датаграмму обработчику IP (сетевой уровень);

– обработчик IP рассматривает переданную UDP датаграмму как данные и предваряет их своим заголовком (в котором, в частности, находится IP-адрес получателя, взятый из той же структуры данных приложения, и номер верхнего протокола);

– полученный пакет обработчик IP передаёт на канальный уровень, который опять-таки рассматривает данный пакет как «сырые» данные;

– обработчик канального уровня, аналогично предыдущим обработчикам, добавляет в начало свой заголовок (в котором так же указывается номер протокола верхнего уровня, в нашем случае это 0x0800(IP)) и, в большинстве случаев, добавляет конечную контрольную сумму, тем самым формируя кадр;

– далее полученный кадр передаётся на физический уровень, который осуществляет преобразование битов в электрические или оптические сигналы и посылает их в среду передачи.

То есть, говоря более простым языком, инкапсуляция — упаковка пакетов (возможно, разного протокола) в пакеты одного протокола, включая адрес.

Типичная структура пакета:

– *стартовая комбинация*, или *преамбула*, которая обеспечивает настройку аппаратуры адаптера или другого сетевого устройства на прием и обработку пакета. Это поле может отсутствовать или сводиться к одному-единственному стартовому биту.

– *сетевой адрес (идентификатор) принимающего абонента*, то есть индивидуальный или групповой номер, присвоенный каждому принимающему абоненту в сети. Этот адрес позволяет приемнику распознать пакет, адресованный ему лично, группе, в которую он входит, или всем абонентам сети одновременно.

– *сетевой адрес (идентификатор) передающего абонента*, то есть индивидуальный или групповой номер, присвоенный каждому передающему

абоненту. Этот адрес информирует принимающего абонента, откуда пришел данный пакет. Включение в пакет адреса передатчика необходимо в том случае, когда одному приемнику могут попеременно приходить пакеты от разных передатчиков.

– *служебная информация*, которая указывает на тип пакета, его номер, размер, формат, маршрут его доставки, на то, что с ним надо делать приемнику и т.д.

– *данные* - та информация, ради передачи которой используется данный пакет. Правда, существуют специальные управляющие пакеты, которые не имеют поля данных. Их можно рассматривать как сетевые команды. Пакеты, включающие поле данных, называются информационными пакетами. Управляющие пакеты могут выполнять функцию начала сеанса связи, конца сеанса связи, подтверждения приема информационного пакета, запроса информационного пакета и т.д.

– *контрольная сумма пакета* - это числовой код, формируемый передатчиком по определенным правилам и содержащий в свернутом виде информацию обо всем пакете. Приемник, повторяя вычисления, сделанные передатчиком, с принятым пакетом, сравнивает их результат с контрольной суммой и делает вывод о правильности или ошибочности передачи пакета. Если пакет ошибочен, то приемник запрашивает его повторную передачу.

– стоповая комбинация служит для информирования аппаратуры принимающего абонента об окончании пакета, обеспечивает выход аппаратуры приемника из состояния приема. Это поле может отсутствовать, если используется самосинхронизирующийся код, позволяющий детектировать факт передачи пакета.

– В сетях с коммутацией пакетов сегодня применяется два класса механизмов передачи пакетов:

– дейтаграммная передача;

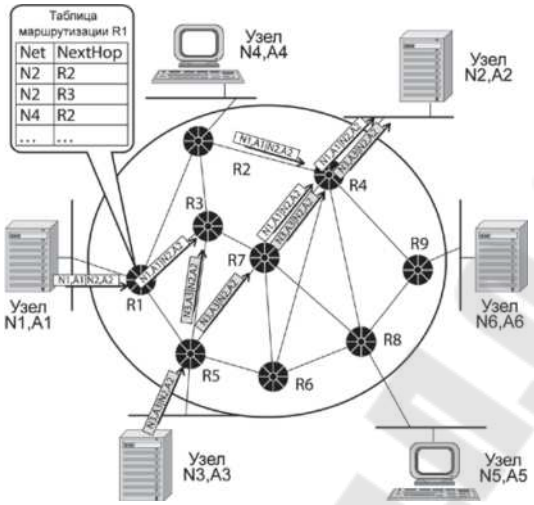
– виртуальные каналы.

– Примерами сетей, реализующих дейтаграммный механизм передачи, являются сети Ethernet, IP и IPX. С помощью виртуальных каналов передают данные сети X.25, frame relay и ATM. Сначала мы рассмотрим базовые принципы дейтаграммного подхода.

– Дейтаграммный способ передачи данных основан на том, что все передаваемые пакеты обрабатываются независимо друг от друга, пакет за пакетом. Принадлежность пакета к определенному потоку между двумя конечными узлами и двумя приложениями, работающими на этих узлах, никак не учитывается.

– Выбор следующего узла — например, коммутатора Ethernet или маршрутизатора IP/IPX — происходит только на основании адреса узла назначения, содержащегося в заголовке пакета. Решение о том, какому узлу передать пришедший пакет, принимается на основе таблицы, содержащей набор адресов назначения и адресную информацию, однозначно определяющую следующий (транзитный или конечный) узел. Такие таблицы имеют разные названия — например, для сетей Ethernet они обычно называются таблицей продвижения

(forwarding table), а для сетевых протоколов, таких как IP и IPX, — таблицами маршрутизации (routing table). Далее для простоты будем пользоваться термином "таблица маршрутизации" в качестве обобщенного названия такого рода таблиц, используемых для дейтаграммной передачи на основании только адреса назначения конечного узла.



– Рисунок 14 – Дейтаграммный принцип передачи пакетов

– В таблице маршрутизации для одного и того же адреса назначения может содержаться несколько записей, указывающих, соответственно, на различные адреса следующего маршрутизатора. Такой подход используется для повышения производительности и надежности сети. В примере на рис. 14 пакеты, поступающие в маршрутизатор R1 для узла назначения с адресом N2, A2, в целях баланса нагрузки распределяются между двумя следующими маршрутизаторами — R2 и R3, что снижает нагрузку на каждый из них, а значит, уменьшает очереди и ускоряет доставку. Некоторая "размытость" путей следования пакетов с одним и тем же адресом назначения через сеть является прямым следствием принципа независимой обработки каждого пакета, присущего дейтаграммным протоколам. Пакеты, следующие по одному и тому же адресу назначения, могут добираться до него разными путями и вследствие изменения состояния сети, например отказа промежуточных маршрутизаторов.

– Такая особенность дейтаграммного механизма как размытость путей следования трафика через сеть также в некоторых случаях является недостатком. Например, если пакетам определенной сессии между двумя конечными узлами сети необходимо обеспечить заданное качество обслуживания. Современные методы поддержки QoS работают эффективней, когда трафик, которому нужно обеспечить гарантии обслуживания, всегда проходит через одни и те же промежуточные узлы.

– Виртуальные каналы в сетях с коммутацией пакетов

– Механизм виртуальных каналов (virtual circuit или virtual channel) создает в сети устойчивые пути следования трафика через сеть с коммутацией пакетов. Этот механизм учитывает существование в сети потоков данных.

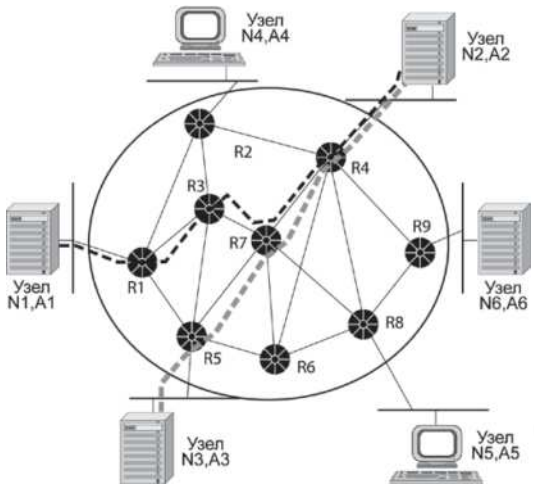


Рисунок 15 – Принцип работы виртуального канала.

Если целью является прокладка для всех пакетов потока единого пути через сеть, то необходимым (но не всегда единственным) признаком такого потока должно быть наличие для всех его пакетов общих точек входа и выхода из сети. Именно для передачи таких потоков в сети создаются виртуальные каналы. На рисунке 15 показан фрагмент сети, в которой проложены два виртуальных канала. Первый проходит от конечного узла с адресом N1, A1 до конечного узла с адресом N2, A2 через промежуточные коммутаторы сети R1, R3, R7 и R4. Второй обеспечивает продвижение данных по пути N3, A3 — R5 — R7 — R4 — N2, A2. Между двумя конечными узлами может быть проложено несколько виртуальных каналов, как полностью совпадающих в отношении пути следования через транзитные узлы, так и отличающихся.

Сеть только обеспечивает возможность передачи трафика вдоль виртуального канала, а какие именно потоки будут передаваться по этим каналам, решают сами конечные узлы. Узел может использовать один и тот же виртуальный канал для передачи всех потоков, которые имеют общие с данным виртуальным каналом конечные точки, или же только части из них. Например, для потока реального времени можно использовать один виртуальный канал, а для трафика электронной почты — другой. В последнем случае разные виртуальные каналы будут предъявлять разные требования к качеству обслуживания, и удовлетворить их будет проще, чем в том случае, когда по одному виртуальному каналу передается трафик с разными требованиями к параметрам QoS.

Важной особенностью сетей с виртуальными каналами является использование локальных адресов пакетов при принятии решения о передаче. Вместо достаточно длинного адреса узла назначения (его длина должна позволять уникально идентифицировать все узлы и подсети в сети, например технология ATM оперирует адресами длиной в 20 байт) применяется локальная, то есть меняющаяся от узла к узлу, метка, которой помечаются все пакеты, перемещаемые по определенному виртуальному каналу. Эта метка в различных технологиях называется по-разному: в технологии X.25 — номер логического канала (Logical Channel number, LCN), в технологии frame relay — идентификатор соединения

уровня канала данных (Data Link Connection Identifier, DLCI), в технологии ATM — идентификатор виртуального канала (Virtual Channel Identifier, VCI). Однако назначение ее везде одинаково — промежуточный узел, называемый в этих технологиях коммутатором, читает значение метки из заголовка пришедшего пакета и просматривает свою таблицу коммутации, в которой указывается, на какой выходной порт нужно передать пакет. Таблица коммутации содержит записи только о проходящих через данный коммутатор виртуальных каналах, а не обо всех имеющихся в сети узлах (или подсетях, если применяется иерархический способ адресации). Обычно в крупной сети количество проложенных через узел виртуальных каналов существенно меньше количества узлов и подсетей, поэтому по размерам таблица коммутации намного меньше таблицы маршрутизации, а, следовательно, просмотр занимает гораздо меньше времени и не требует от коммутатора большой вычислительной мощности.

Идентификатор виртуального канала (именно такое название метки будет использоваться далее) также намного короче адреса конечного узла (по той же причине), поэтому и избыточность заголовка пакета, который теперь не содержит длинного адреса, а переносит по сети только идентификатор, существенно меньше.

Виртуальный путь – это соединение между двумя коммутаторами сети ATM, описанные в таблицах коммутации соответствующих коммутаторов. Виртуальные пути применяются для наиболее часто используемых направлений. По одному виртуальному пути могут передаваться несколько виртуальных каналов. Виртуальный путь существует независимо от того, идет по нему передача данных или нет. Всего виртуальных путей в рамках сети может быть 256. В каждом виртуальном пути м.б. до 65 000 соединений.

Виртуальный канал – это соединение между двумя конечными станциями сети ATM. Виртуальный канал является двунаправленным.

Имеются три вида виртуальных каналов:

1) постоянные виртуальные каналы (PVC). PVC устанавливается вручную в процессе конфигурирования сети.

2) коммутируемые виртуальные каналы (SVC). SVC устанавливается по мере необходимости всякий раз, когда конечная станция пытается передать данные другой станции. Это наиболее часто используемый тип каналов.

3) интеллектуальные постоянные виртуальные каналы (SPVC). SPVC представляет собой гибрид двух предыдущих типов каналов. Данное соединение устанавливается вручную на этапе конфигурирования сети, однако провайдер ATM знает только конечные станции.

Преимущества PVC:

1) не тратится время на установление соединения, поэтому обеспечивается более высокая производительность сети.

2) обеспечивается лучший контроль над сетью.

Недостаток: они должны формироваться вручную

Преимущества SVC:

1) данный вид соединения лучше установить или устранить, нежели PVC.

2) с помощью SVC могут эмулироваться каналы без установления соединения.

3) SVC требует меньше затрат на обслуживание, т.к. данное соединение проводится автоматически, а не вручную.

4) данный вид соединения имеет более высокую отказоустойчивость.

Преимущества SPVC:

1) Позволяет заранее задать конечные станции, поэтому не приходится тратить время на установление соединения.

2) Имеет более высокую отказоустойчивость подобно SVC.

Тема 11. Коммутация каналов, коммутация пакетов, коммутация сообщений.

Разные подходы к выполнению коммутации

В общем случае решение каждой из частных задач коммутации — определение потоков и соответствующих маршрутов, фиксация маршрутов в конфигурационных параметрах и таблицах сетевых устройств, распознавание потоков и передача данных между интерфейсами одного устройства, мультиплексирование/демультиплексирование потоков и разделение среды передачи — тесно связано с решением всех остальных. Комплекс технических решений обобщенной задачи коммутации в совокупности составляет базис любой сетевой технологии. От того, какой механизм прокладки маршрутов, продвижения данных и совместного использования каналов связи заложен в той или иной сетевой технологии, зависят ее фундаментальные свойства.

Среди множества возможных подходов к решению задачи коммутации абонентов в сетях выделяют два основополагающих:

коммутация каналов (circuit switching);

коммутация пакетов (packet switching).

Внешне обе эти схемы соответствуют приведенной на рис. 16 структуре сети, однако возможности и свойства их различны.

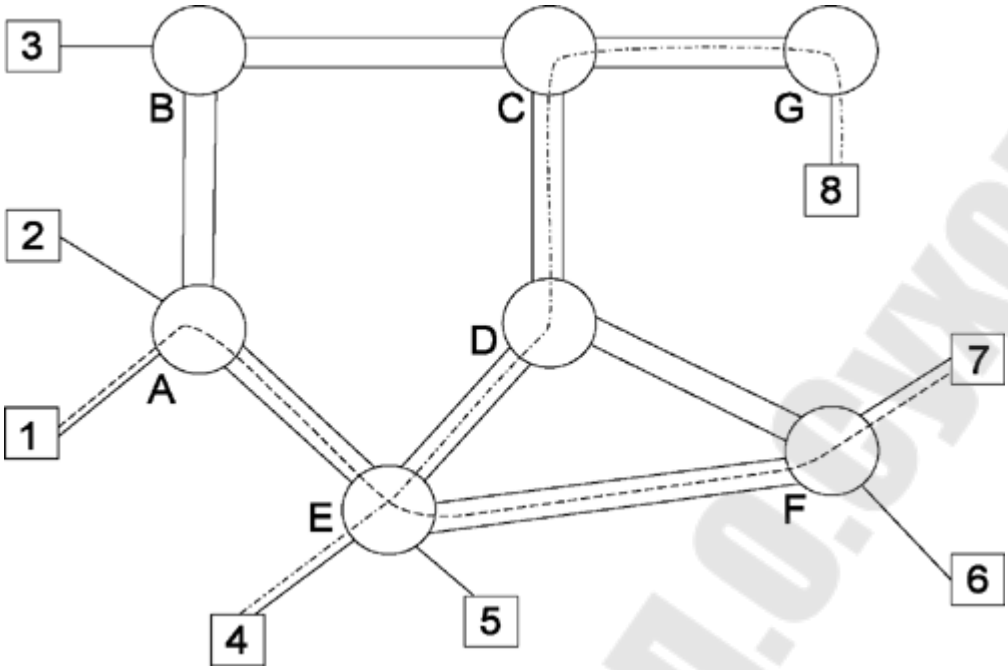


Рисунок 16 – Общая структура сети с коммутацией абонентов

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

Коммутация каналов

При коммутации каналов коммутационная сеть образует между конечными узлами непрерывный составной физический канал из последовательно соединенных коммутаторами промежуточных канальных участков. Условием того, что несколько физических каналов при последовательном соединении образуют единый физический канал, является равенство скоростей передачи данных в каждом из составляющих физических каналов. Равенство скоростей означает, что коммутаторы такой сети не должны буферизовать передаваемые данные.

В сети с коммутацией каналов перед передачей данных всегда необходимо выполнить процедуру установления соединения, в процессе которой и создается составной канал. И только после этого можно начинать передавать данные.

Например, если сеть, изображенная на рис. 16, работает по технологии коммутации каналов, то узел 1, чтобы передать данные узлу 7, сначала должен передать специальный запрос на установление соединения коммутатору А, указав адрес назначения 7. Коммутатор А должен выбрать маршрут образования составного канала, а затем передать запрос следующему коммутатору, в данном случае Е. Затем коммутатор Е передает запрос коммутатору F, а тот, в свою очередь, передает запрос узлу 7. Если узел 7 принимает запрос на установление соединения, он направляет по уже установленному каналу ответ исходному узлу, после чего

составной канал считается скоммутированным, и узлы 1 и 7 могут обмениваться по нему данными.

Техника коммутации каналов имеет свои достоинства и недостатки.

Достоинства коммутации каналов

Постоянная и известная скорость передачи данных по установленному между конечными узлами каналу. Это дает пользователю сети возможности на основе заранее произведенной оценки необходимой для качественной передачи данных пропускной способности установить в сети канал нужной скорости.

Низкий и постоянный уровень задержки передачи данных через сеть. Это позволяет качественно передавать данные, чувствительные к задержкам (называемые также трафиком реального времени) — голос, видео, различную технологическую информацию.

Недостатки коммутации каналов

Отказ сети в обслуживании запроса на установление соединения. Такая ситуация может сложиться из-за того, что на некотором участке сети соединение нужно установить вдоль канала, через который уже проходит максимально возможное количество информационных потоков. Отказ может случиться и на конечном участке составного канала — например, если абонент способен поддерживать только одно соединение, что характерно для многих телефонных сетей. При поступлении второго вызова к уже разговаривающему абоненту сеть передает вызывающему абоненту короткие гудки — сигнал "занято".

Рисунок 17 – Установление составного канала

Нерациональное использование пропускной способности физических каналов. Та часть пропускной способности, которая отводится составному каналу после установления соединения, предоставляется ему на все время, т.е. до тех пор, пока

соединение не будет разорвано. Однако абонентам не всегда нужна пропускная способность канала во время соединения, например в телефонном разговоре могут быть паузы, еще более неравномерным во времени является взаимодействие компьютеров. Невозможность динамического перераспределения пропускной способности представляет собой принципиальное ограничение сети с коммутацией каналов, так как единицей коммутации здесь является информационный поток в целом.

Обязательная задержка перед передачей данных из-за фазы установления соединения.

Достоинства и недостатки любой сетевой технологии относительны. В определенных ситуациях на первый план выходят достоинства, а недостатки становятся несущественными. Так, техника коммутации каналов хорошо работает в тех случаях, когда нужно передавать только трафик телефонных разговоров. Здесь с невозможностью "вырезать" паузы из разговора и более рационально использовать магистральные физические каналы между коммутаторами можно мириться. А вот при передаче очень неравномерного компьютерного трафика эта нерациональность уже выходит на первый план.

Коммутация пакетов

Эта техника коммутации была специально разработана для эффективной передачи компьютерного трафика. Первые шаги на пути создания компьютерных сетей на основе техники коммутации каналов показали, что этот вид коммутации не позволяет достичь высокой общей пропускной способности сети. Типичные сетевые приложения генерируют трафик очень неравномерно, с высоким уровнем пульсации скорости передачи данных. Например, при обращении к удаленному файловому серверу пользователь сначала просматривает содержимое каталога этого сервера, что порождает передачу небольшого объема данных. Затем он открывает требуемый файл в текстовом редакторе, и эта операция может создать достаточно интенсивный обмен данными, особенно если файл содержит объемные графические включения. После отображения нескольких страниц файла пользователь некоторое время работает с ними локально, что вообще не требует передачи данных по сети, а затем возвращает модифицированные копии страниц на сервер — и это снова порождает интенсивную передачу данных по сети.

Коэффициент пульсации трафика отдельного пользователя сети, равный отношению средней интенсивности обмена данными к максимально возможной, может достигать 1:50 или даже 1:100. Если для описанной сессии организовать коммутацию канала между компьютером пользователя и сервером, то большую часть времени канал будет простаивать. В то же время коммутационные возможности сети будут закреплены за данной парой абонентов и будут недоступны другим пользователям сети.

При коммутации пакетов все передаваемые пользователем сообщения разбиваются в исходном узле на сравнительно небольшие части, называемые

пакетами. Напомним, что сообщением называется логически завершенная порция данных — запрос на передачу файла, ответ на этот запрос, содержащий весь файл и т.д. Сообщения могут иметь произвольную длину, от нескольких байт до многих мегабайт. Напротив, пакеты обычно тоже могут иметь переменную длину, но в узких пределах, например от 46 до 1500 байт. Каждый пакет снабжается заголовком, в котором указывается адресная информация, необходимая для доставки пакета на узел назначения, а также номер пакета, который будет использоваться узлом назначения для сборки сообщения (рис. 18). Пакеты транспортируются по сети как независимые информационные блоки. Коммутаторы сети принимают пакеты от конечных узлов и на основании адресной информации передают их друг другу, а в конечном итоге — узлу назначения.

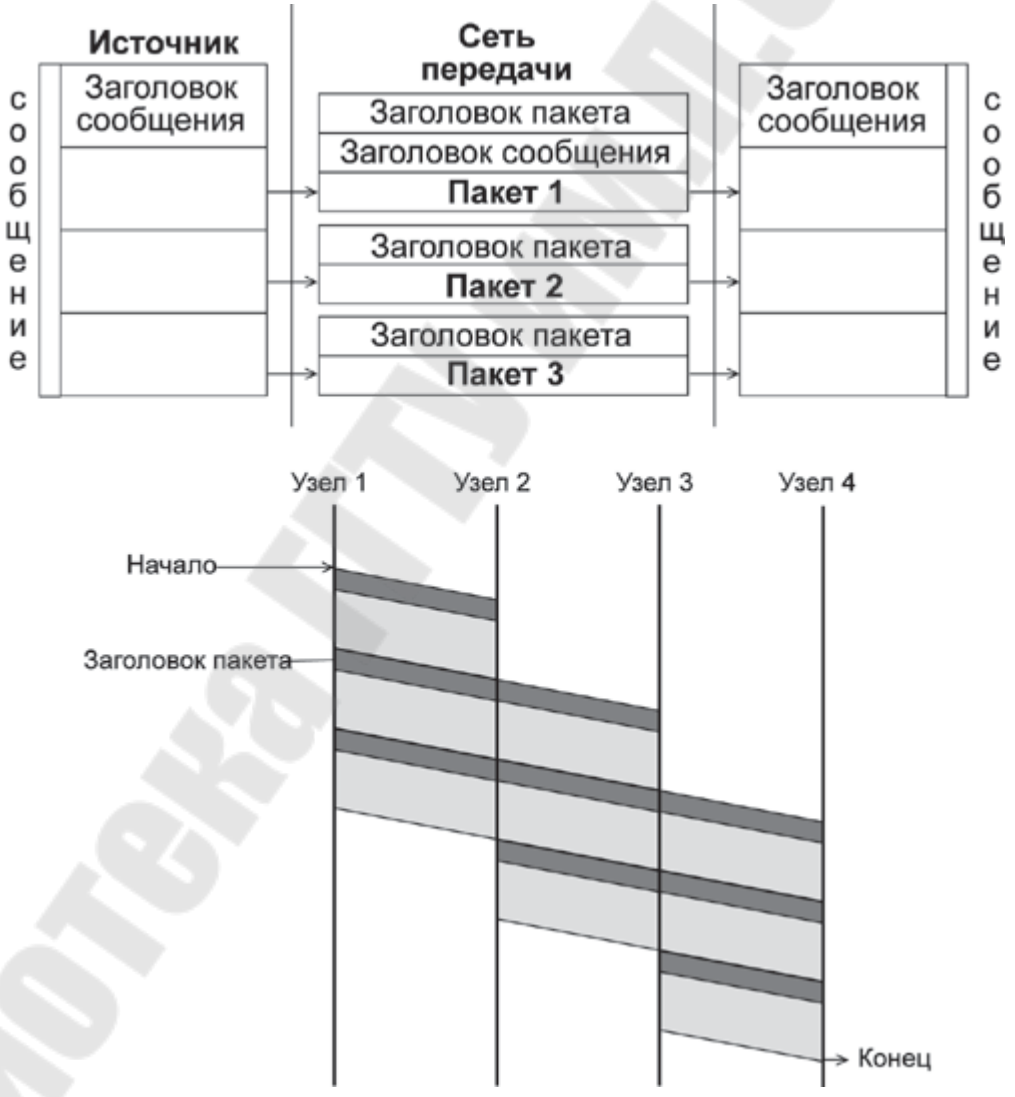


Рисунок 18 – Разбиение сообщения на пакеты

Коммутаторы пакетной сети отличаются от коммутаторов каналов тем, что они имеют внутреннюю буферную память для временного хранения пакетов, если выходной порт коммутатора в момент принятия пакета занят передачей другого пакета (рис. 18). В этом случае пакет находится некоторое время в очереди пакетов в буферной памяти выходного порта, а когда до него дойдет очередь, он передается следующему коммутатору. Такая схема передачи данных позволяет сглаживать

пульсацию трафика на магистральных связях между коммутаторами и тем самым наиболее эффективно использовать их для повышения пропускной способности сети в целом.

Действительно, для пары абонентов наиболее эффективным было бы предоставление им в единоличное пользование скоммутированного канала связи, как это делается в сетях с коммутацией каналов. В таком случае время взаимодействия этой пары абонентов было бы минимальным, так как данные без задержек передавались бы от одного абонента другому. Простои канала во время пауз передачи абонентов не интересуют, для них важно быстрее решить свою задачу. Сеть с коммутацией пакетов замедляет процесс взаимодействия конкретной пары абонентов, так как их пакеты могут ожидать в коммутаторах, пока по магистральным связям передаются другие пакеты, пришедшие в коммутатор ранее.

Тем не менее, общий объем передаваемых сетью компьютерных данных в единицу времени при технике коммутации пакетов будет выше, чем при технике коммутации каналов. Это происходит потому, что пульсации отдельных абонентов в соответствии с законом больших чисел распределяются во времени так, что их пики не совпадают. Поэтому коммутаторы постоянно и достаточно равномерно загружены работой, если число обслуживаемых ими абонентов действительно велико. На рис. 19 показано, что трафик, поступающий от конечных узлов на коммутаторы, распределен во времени очень неравномерно. Однако коммутаторы более высокого уровня иерархии, которые обслуживают соединения между коммутаторами нижнего уровня, загружены более равномерно, и поток пакетов в магистральных каналах, соединяющих коммутаторы верхнего уровня, имеет почти максимальный коэффициент использования. Буферизация сглаживает пульсации, поэтому коэффициент пульсации на магистральных каналах гораздо ниже, чем на каналах абонентского доступа — он может быть равным 1:10 или даже 1:2.

Более высокая эффективность сетей с коммутацией пакетов по сравнению с сетями с коммутацией каналов (при равной пропускной способности каналов связи) была доказана в 60-е годы как экспериментально, так и с помощью имитационного моделирования. Здесь уместна аналогия с мультипрограммными операционными системами. Каждая отдельная программа в такой системе выполняется дольше, чем в однопрограммной системе, когда программе выделяется все процессорное время, пока ее выполнение не завершится. Однако общее число программ, выполняемых за единицу времени, в мультипрограммной системе больше, чем в однопрограммной.

Сеть с коммутацией пакетов замедляет процесс взаимодействия конкретной пары абонентов, но повышает пропускную способность сети в целом.

Задержки в источнике передачи:

- время на передачу заголовков;
- задержки, вызванные интервалами между передачей каждого следующего пакета.

Рисунок 19 – Сглаживание пульсаций трафика в сети с коммутацией пакетов

Задержки в каждом коммутаторе:

– время буферизации пакета;

– время коммутации, которое складывается из:

– времени ожидания пакета в очереди (переменная величина);

– времени перемещения пакета в выходной порт.

Достоинства коммутации пакетов

1. Высокая общая пропускная способность сети при передаче пульсирующего трафика.

2. Возможность динамически перераспределять пропускную способность физических каналов связи между абонентами в соответствии с реальными потребностями их трафика.

3. Недостатки коммутации пакетов

4. Неопределенность скорости передачи данных между абонентами сети, обусловленная тем, что задержки в очередях буферов коммутаторов сети зависят от общей загрузки сети.

5. Переменная величина задержки пакетов данных, которая может быть достаточно продолжительной в моменты мгновенных перегрузок сети.

6. Возможные потери данных из-за переполнения буферов.

В настоящее время активно разрабатываются и внедряются методы, позволяющие преодолеть указанные недостатки, которые особенно остро проявляются для чувствительного к задержкам трафика, требующего при этом постоянной скорости передачи. Такие методы называются методами обеспечения качества обслуживания (Quality of Service, QoS).

Сети с коммутацией пакетов, в которых реализованы методы обеспечения качества обслуживания, позволяют одновременно передавать различные виды

трафика, в том числе такие важные как телефонный и компьютерный. Поэтому методы коммутации пакетов сегодня считаются наиболее перспективными для построения конвергентной сети, которая обеспечит комплексные качественные услуги для абонентов любого типа. Тем не менее, нельзя сбрасывать со счетов и методы коммутации каналов. Сегодня они не только с успехом работают в традиционных телефонных сетях, но и широко применяются для образования высокоскоростных постоянных соединений в так называемых первичных (опорных) сетях технологий SDH и DWDM, которые используются для создания магистральных физических каналов между коммутаторами телефонных или компьютерных сетей. В будущем вполне возможно появление новых технологий коммутации, в том или ином виде комбинирующих принципы коммутации пакетов и каналов.

Коммутация сообщений

Коммутация сообщений по своим принципам близка к коммутации пакетов. Под коммутацией сообщений понимается передача единого блока данных между транзитными компьютерами сети с временной буферизацией этого блока на диске каждого компьютера. Сообщение в отличие от пакета имеет произвольную длину, которая определяется не технологическими соображениями, а содержанием информации, составляющей сообщение.

Транзитные компьютеры могут соединяться между собой как сетью с коммутацией пакетов, так и сетью с коммутацией каналов. Сообщение (это может быть, например, текстовый документ, файл с кодом программы, электронное письмо) хранится в транзитном компьютере на диске, причем довольно продолжительное время, если компьютер занят другой работой или сеть временно перегружена.

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

Режим коммутации сообщений разгружает сеть для передачи трафика, требующего быстрого ответа, например трафика службы WWW или файловой службы.

Количество транзитных компьютеров обычно стараются уменьшить. Если компьютеры подключены к сети с коммутацией пакетов, то число промежуточных компьютеров уменьшается до двух. Например, пользователь передает почтовое сообщение своему серверу исходящей почты, а тот сразу старается передать его серверу входящей почты адресата. Но если компьютеры связаны между собой телефонной сетью, то часто используется несколько промежуточных серверов, так как прямой доступ к конечному серверу может быть в данный момент невозможен из-за перегрузки телефонной сети (абонент занят) или экономически невыгоден из-за высоких тарифов на дальнюю телефонную связь.

Техника коммутации сообщений появилась в компьютерных сетях раньше техники коммутации пакетов, но потом была вытеснена последней, как более эффективной по критерию пропускной способности сети. Запись сообщения на диск занимает достаточно много времени, и кроме того, наличие дисков предполагает использование в качестве коммутаторов специализированных компьютеров, что влечет за собой существенные затраты на организацию сети.

Сегодня коммутация сообщений работает только для некоторых не оперативных служб, причем чаще всего поверх сети с коммутацией пакетов, как служба прикладного уровня.

Сравнение способов коммутации

Сравнение коммутации каналов и коммутации пакетов

Сравнение коммутации каналов и коммутации пакетов	
Коммутация каналов	**Коммутация пакетов**
Гарантированная пропускная способность (полоса) для взаимодействующих абонентов	Пропускная способность сети для абонентов неизвестна, задержки передачи носят случайный характер
Сеть может отказать абоненту в установлении соединения	Сеть всегда готова принять данные от абонента
Трафик реального времени передается без задержек	Ресурсы сети используются эффективно при передаче пульсирующего трафика
Адрес используется только на этапе установления соединения	Адрес передается с каждым пакетом

Постоянная и динамическая коммутация

Как сети с коммутацией пакетов, так и сети с коммутацией каналов можно разделить на два класса:

– сети с динамической коммутацией;

– сети с постоянной коммутацией.

В сетях с динамической коммутацией:

– разрешается устанавливать соединение по инициативе пользователя сети;

– коммутация выполняется только на время сеанса связи, а затем (по инициативе одного из пользователей) разрывается;

– в общем случае пользователь сети может соединиться с любым другим пользователем сети;

– время соединения между парой пользователей при динамической коммутации составляет от нескольких секунд до нескольких часов и завершается после выполнения определенной работы — передачи файла, просмотра страницы текста или изображения и т.п.

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

Сеть, работающая в режиме постоянной коммутации:

– разрешает паре пользователей заказать соединение на длительный период времени;

– соединение устанавливается не пользователями, а персоналом, обслуживающим сеть;

– период, на который устанавливается постоянная коммутация, составляет обычно несколько месяцев;

– режим постоянной (permanent) коммутации в сетях с коммутацией каналов часто называется сервисом выделенных (dedicated) или арендуемых (leased) каналов;

– в том случае, когда постоянное соединение через сеть коммутаторов устанавливается с помощью автоматических процедур, инициированных обслуживающим персоналом, его часто называют полупостоянным (semi-permanent) соединением, в отличие от режима ручного конфигурирования каждого коммутатора.

Наиболее популярными сетями, работающими в режиме постоянной коммутации, сегодня являются сети технологии SDH, на основе которых строятся выделенные каналы связи с пропускной способностью в несколько гигабит в секунду.

Некоторые типы сетей поддерживают оба режима работы. Например, сети X.25 и ATM могут предоставлять пользователю возможность динамически связаться с любым другим пользователем сети и в то же время отправлять данные по постоянному соединению определенному абоненту.

Пропускная способность сетей с коммутацией пакетов

Одним из отличий метода коммутации пакетов от метода коммутации каналов является неопределенность пропускной способности соединения между двумя абонентами. В случае коммутации каналов после образования составного канала пропускная способность сети при передаче данных между конечными узлами известна — это пропускная способность канала. Данные после задержки, связанной с установлением канала, начинают передаваться на максимальной для канала скорости (рис. 20).

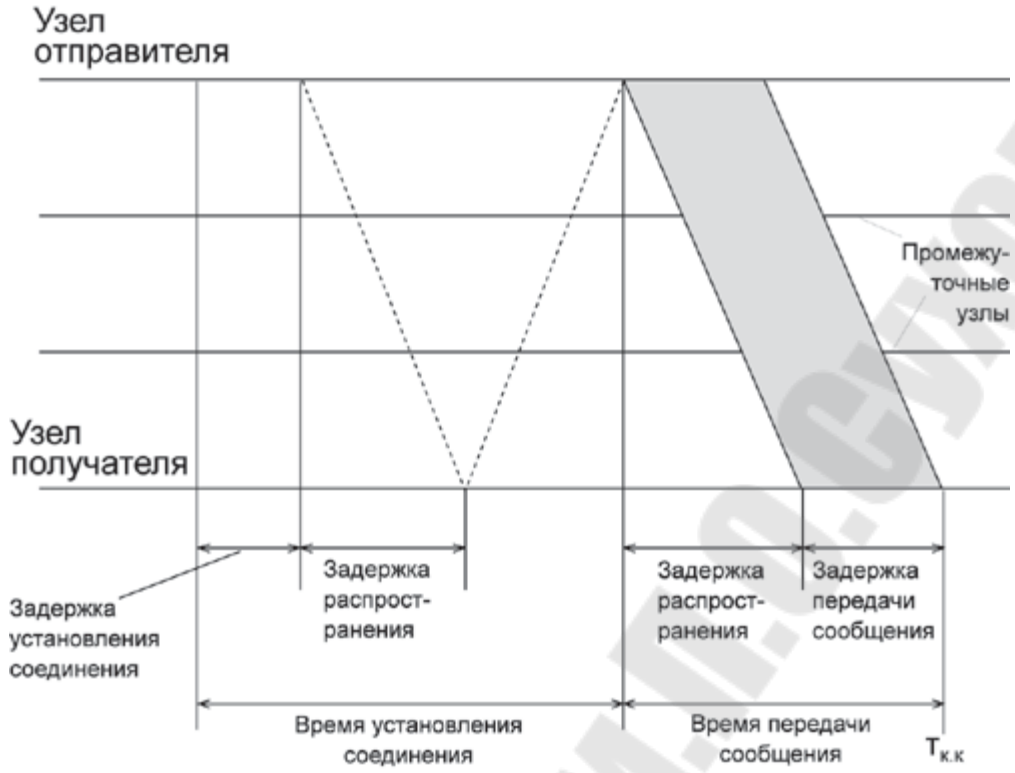


Рисунок 20 – Задержки передачи данных в сетях с коммутацией каналов.

Время передачи сообщения в сети с коммутацией каналов Тк.к. равно сумме задержки распространения сигнала по линии связи и задержки передачи сообщения. Задержка распространения сигнала зависит от скорости распространения электромагнитных волн в конкретной физической среде, которая колеблется от 0,6 до 0,9 скорости света в вакууме. Время передачи сообщения равно V/C, где V — объем сообщения в битах, а C — пропускная способность канала в битах в секунду.

В сети с коммутацией пакетов картина совсем иная.

Процедура установления соединения в этих сетях, если она используется, занимает примерно такое же время, как и в сетях с коммутацией каналов, поэтому будем сравнивать только время передачи данных.

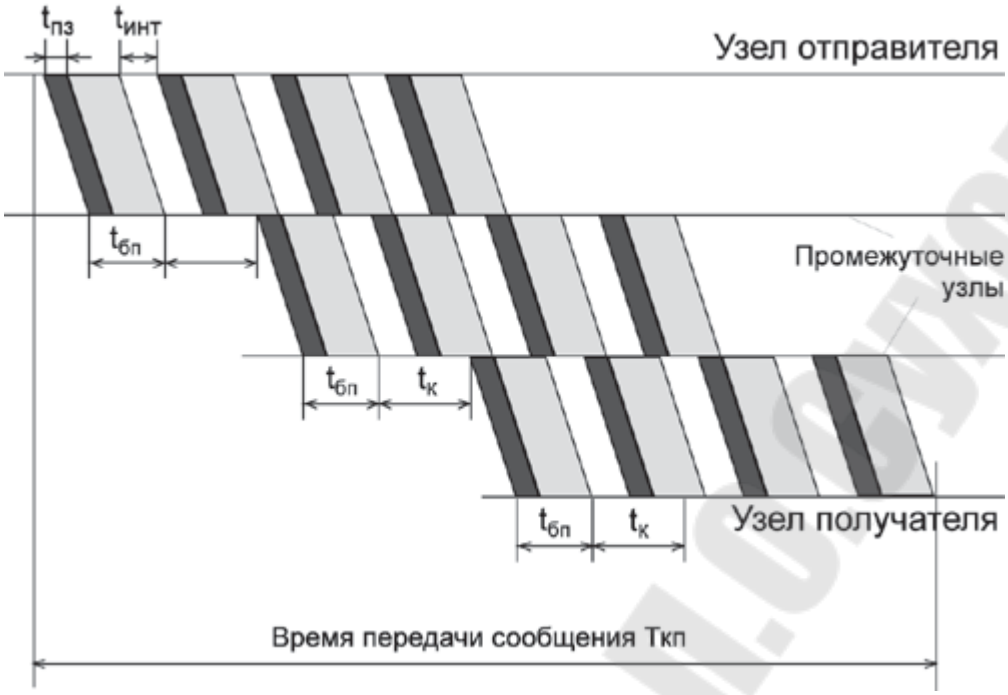


Рисунок 21 – Задержки при передаче данных в сетях с коммутацией пакетов.

На рис. 21 показан пример передачи данных в сети с коммутацией пакетов. Предполагается, что по сети передается сообщение того же объема, что и сообщение, передаваемое на рис. 20 однако оно разделено на пакеты, каждый из которых снабжен заголовком. Время передачи сообщения в сети с коммутацией пакетов обозначено на рисунке Тк.п. При передаче этого разбитого на пакеты сообщения по сети с коммутацией пакетов возникают дополнительные задержки. Во-первых, это задержки в источнике передачи, который, помимо передачи собственно сообщения, тратит дополнительное время на передачу заголовков tп.з., к тому же добавляются задержки tинт, вызванные интервалами между передачей каждого следующего пакета (это время уходит на формирование очередного пакета стеком протоколов).

Во-вторых, дополнительное время тратится в каждом коммутаторе. Здесь задержки складываются из времени буферизации пакета tб.п. (коммутатор не может начать передачу пакета, не приняв его полностью в свой буфер) и времени коммутации tк. Время буферизации равно времени приема пакета с битовой скоростью протокола. Время коммутации складывается из времени ожидания пакета в очереди и времени перемещения пакета в выходной порт. Если время перемещения пакета фиксировано и, как правило, невелико (от нескольких микросекунд до нескольких десятков микросекунд), то время ожидания пакета в очереди колеблется в очень широких пределах и заранее неизвестно, так как зависит от текущей загрузки сети.

Проведем грубую оценку задержки при передаче данных в сетях с коммутацией пакетов по сравнению с сетями с коммутацией каналов на простейшем примере. Пусть тестовое сообщение, которое нужно передать в обоих видах сетей,

имеет объем 200 Кбайт. Отправитель находится от получателя на расстоянии 5000 км. Пропускная способность линий связи составляет 2 Мбит/с.

Время передачи данных по сети с коммутацией каналов складывается из времени распространения сигнала, которое для расстояния 5000 км можно оценить примерно в 25 мс (принимая скорость распространения сигнала равной 2/3 скорости света), и времени передачи сообщения, которое при пропускной способности 2 Мбит/с и длине сообщения 200 Кбайт равно примерно 800 мс. При расчете корректное значение К (210), равное 1024, округлялось до 1000, аналогично значение М (220), равное 1048576, округлялось до 1000000. Таким образом, передача данных оценивается в 825 мс.

Ясно, что при передаче этого сообщения по сети с коммутацией пакетов, обладающей такой же суммарной длиной и пропускной способностью каналов, пролегающих от отправителя к получателю, время распространения сигнала и время передачи данных будут такими же — 825 мс. Однако из-за задержек в промежуточных узлах общее время передачи данных увеличится. Давайте оценим, на сколько возрастет это время. Будем считать, что путь от отправителя до получателя пролегает через 10 коммутаторов. Пусть исходное сообщение разбивается на пакеты в 1 Кбайт, всего 200 пакетов. Вначале оценим задержку, которая возникает в исходном узле. Предположим, что доля служебной информации, размещенной в заголовках пакетов, по отношению к общему объему сообщения составляет 10%. Следовательно, дополнительная задержка, связанная с передачей заголовков пакетов, составляет 10% от времени передачи целого сообщения, то есть 80 мс. Если принять интервал между отправкой пакетов равным 1 мс, то дополнительные потери за счет интервалов составят 200 мс. Таким образом, в исходном узле из-за пакетирования сообщения при передаче возникла дополнительная задержка в 280 мс.

Каждый из 10 коммутаторов вносит задержку коммутации, которая может составлять от долей до тысяч миллисекунд. В данном примере будем считать, что на коммутацию в среднем тратится 20 мс. Кроме того, при прохождении сообщений через коммутатор возникает задержка буферизации пакета. Эта задержка при величине пакета 1 Кбайт и пропускной способности линии 2 Мбит/с равна 4 мс. Общая задержка, вносимая 10 коммутаторами, составляет примерно 240 мс. В результате дополнительная задержка, созданная сетью с коммутацией пакетов, составила 520 мс. Учитывая, что вся передача данных в сети с коммутацией каналов заняла 825 мс, эту дополнительную задержку можно считать существенной.

Хотя приведенный расчет носит очень приблизительный характер, он объясняет, почему процесс передачи для определенной пары абонентов в сети с коммутацией пакетов является более медленным, чем в сети с коммутацией каналов.

Неопределенная пропускная способность сети с коммутацией пакетов — это плата за ее общую эффективность при некотором ущемлении интересов отдельных абонентов. Аналогично, в мультипрограммной операционной системе время

выполнения приложения предсказать невозможно, так как оно зависит от количества других приложений, с которыми данное приложение делит процессор.

На эффективность работы сети влияют размеры пакетов, которые передает сеть. Слишком большие размеры пакетов приближают сеть с коммутацией пакетов к сети с коммутацией каналов, поэтому эффективность сети падает. Кроме того, при большом размере пакетов увеличивается время буферизации на каждом коммутаторе. Слишком маленькие пакеты заметно увеличивают долю служебной информации, так как каждый пакет содержит заголовок фиксированной длины, а количество пакетов, на которые разбиваются сообщения, при уменьшении размера пакета будет резко расти. Существует некоторая "золотая середина", когда обеспечивается максимальная эффективность работы сети, однако это соотношение трудно определить точно, так как оно зависит от многих факторов, в том числе изменяющихся в процессе работы сети. Поэтому разработчики протоколов для сетей с коммутацией пакетов выбирают пределы, в которых может находиться размер пакета, а точнее его поле данных, так как заголовок, как правило, имеет фиксированную длину. Обычно нижний предел поля данных выбирается равным нулю, что дает возможность передавать служебные пакеты без пользовательских данных, а верхний предел не превышает 4 Кбайт. Приложения при передаче данных пытаются занять максимальный размер поля данных, чтобы быстрее выполнить обмен, а небольшие пакеты обычно используются для коротких служебных сообщений, содержащих, к примеру, подтверждение доставки пакета.

При выборе размера пакета необходимо также учитывать интенсивность битовых ошибок канала. На ненадежных каналах необходимо уменьшать размеры пакетов, так как это сокращает объем повторно передаваемых данных при искажениях пакетов.

Ethernet — пример стандартной технологии коммутации пакетов

Рассмотрим, каким образом описанные выше общие подходы к решению проблем построения сетей воплощены в наиболее популярной сетевой технологии — Ethernet. (Заметим, что мы не будем сейчас подробно рассматривать саму технологию — отложим этот важный вопрос до следующего курса, а сегодня остановимся лишь на некоторых принципиальных моментах, иллюстрирующих ряд уже рассмотренных базовых концепций.)

Сетевая технология — это согласованный набор стандартных протоколов и программно-аппаратных средств (например, сетевых адаптеров, драйверов, кабелей и разъемов), достаточный для построения вычислительной сети.

Эпитет "достаточный" подчеркивает то обстоятельство, что речь идет о минимальном наборе средств, с помощью которых можно построить работоспособную сеть. Эту сеть можно усовершенствовать, например, за счет выделения в ней подсетей, что сразу потребует кроме протоколов стандарта Ethernet применения протокола IP, а также специальных коммуникационных устройств — маршрутизаторов. Усовершенствованная сеть будет, скорее всего, более надежной и

быстродействующей, но за счет надстроек над средствами технологии Ethernet, которая составила базис сети.

Термин "сетевая технология" чаще всего используется в описанном выше узком смысле, но иногда применяется и его расширенное толкование как любого набора средств и правил для построения сети, например "технология сквозной маршрутизации", "технология создания защищенного канала", "технология IP-сетей".

Протоколы, на основе которых строится сеть определенной технологии (в узком смысле), создавались специально для совместной работы, поэтому от разработчика сети не требуется дополнительных усилий по организации их взаимодействия. Иногда сетевые технологии называют базовыми технологиями, имея в виду, что на их основе строится базис любой сети. Примерами базовых сетевых технологий могут служить наряду с Ethernet такие известные технологии локальных сетей как Token Ring и FDDI, или же технологии территориальных сетей X.25 и frame relay. Для получения работоспособной сети в этом случае достаточно приобрести программные и аппаратные средства, относящиеся к одной базовой технологии — сетевые адаптеры с драйверами, концентраторы, коммутаторы, кабельную систему и т. п., — и соединить их в соответствии с требованиями стандарта на данную технологию.

Итак, для сетевой технологии Ethernet характерны:

- коммутация пакетов;
- типовая топология "общая шина";
- плоская числовая адресация;
- разделяемая передающая среда.

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

В стандарте Ethernet строго зафиксирована топология электрических связей. Компьютеры подключаются к разделяемой среде в соответствии с типовой структурой "общая шина" (рис. 22). С помощью разделяемой во времени шины любые два компьютера могут обмениваться данными. Управление доступом к линии связи осуществляется специальными контроллерами — сетевыми адаптерами Ethernet. Каждый компьютер, а точнее, каждый сетевой адаптер, имеет уникальный адрес. Передача данных происходит со скоростью 10 Мбит/с. Эта величина является пропускной способностью сети Ethernet.

Рисунок 22 – Сеть Ethernet.

Суть случайного метода доступа состоит в следующем. Компьютер в сети Ethernet может передавать данные по сети, только если сеть свободна, то есть если никакой другой компьютер в данный момент не занимается обменом. Поэтому важной частью технологии Ethernet является процедура определения доступности среды.

После того как компьютер убедился, что сеть свободна, он начинает передачу и при этом "захватывает" среду. Время монопольного использования разделяемой среды одним узлом ограничивается временем передачи одного кадра. Кадр — это единица данных, которыми обмениваются компьютеры в сети Ethernet. Кадр имеет фиксированный формат и наряду с полем данных содержит различную служебную информацию, например адрес получателя и адрес отправителя.

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

Может возникнуть ситуация, когда несколько компьютеров одновременно решают, что сеть свободна, и начинают передавать информацию. Такая ситуация, называемая коллизией, препятствует правильной передаче данных по сети. В стандарте Ethernet предусмотрен алгоритм обнаружения и корректной обработки коллизий. Вероятность возникновения коллизии зависит от интенсивности сетевого трафика.

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

Основные достоинства технологии Ethernet

Главным достоинством сетей Ethernet, благодаря которому они стали такими популярными, является их экономичность. Для построения сети достаточно иметь по одному сетевому адаптеру для каждого компьютера плюс один физический сегмент коаксиального кабеля нужной длины.

Кроме того, в сетях Ethernet реализованы достаточно простые алгоритмы доступа к среде, адресации и передачи данных. Простота логики работы сети ведет к упрощению и, соответственно, снижению стоимости сетевых адаптеров и их драйверов. По той же причине адаптеры сети Ethernet обладают высокой надежностью.

И, наконец, еще одним замечательным свойством сетей Ethernet является их хорошая расширяемость, то есть возможность подключения новых узлов.

Другие базовые сетевые технологии, такие как Token Ring и FDDI, хотя и обладают индивидуальными чертами, в то же время имеют много общего с Ethernet. В первую очередь, это применение регулярных фиксированных топологий ("иерархическая звезда" и "кольцо"), а также разделяемых сред передачи данных. Существенные отличия одной технологии от другой связаны с особенностями используемого метода доступа к разделяемой среде. Так, отличия технологии Ethernet от технологии Token Ring во многом определяются спецификой заложенных в них методов разделения среды — случайного алгоритма доступа в Ethernet и метода доступа путем передачи маркера в Token Ring.

РАЗДЕЛ 4. СТЕК ПРОТОКОЛОВ TCP/IP

Тема 12. Протокол IP

Ранние эксперименты по передаче и приему информации с помощью компьютеров начались еще в 50-х годах и имели лабораторный характер. Лишь в конце 60-х годов на средства Агентства Перспективных Разработок министерства обороны США была создана **сеть национального масштаба**. Она получила название **ARPANET**. Эта сеть связывала несколько крупных научных, исследовательских и образовательных центров. Ее основной задачей была координация групп коллективов, работающих над едиными научно-техническими проектами, а основным назначением стал обмен электронной почтой файлами с научной и проектно-конструкторской документацией.

Сеть ARPANET заработала в 1969 году. Немногочисленные узлы, входившие в нее в то время, были связаны выделенными линиями. Прием и передача информации обеспечивались программами, работающими на узловых компьютерах. Сеть постепенно расширялась за счет подключения новых узлов, а к началу 80-х годов на базе наиболее крупных узлов были созданы свои региональные сети, воссоздающие общую архитектуру ARPANET на более низком уровне (в региональном или локальном масштабе).

По-настоящему **рождением Интернета** принято считать 1983 год. В этом году произошли революционные изменения в программном обеспечении компьютерной связи. Днем рождения Интернета в современном понимании этого слова стала дата стандартизации протокола связи TCP/IP, лежащего в основе Всемирной сети по нынешний день.

TCP/IP — это не один сетевой протокол, а несколько протоколов, лежащих на разных уровнях сетевой модели OSI (это так называемый стек протоколов). Из них протокол TCP — протокол транспортного уровня. Он управляет тем, как происходит передача информации. Протокол IP— адресный. Он принадлежит сетевому уровню и определяет, куда происходит передача.

Протокол TCP.

Согласно протоколу TCP, отправляемые данные «нарезаются» на небольшие пакеты, после чего каждый пакет маркируется таким образом, чтобы в нем были данные, необходимые для правильной сборки документа на компьютере получателя.

Для понимания сути протокола TCP можно представить игру в шахматы по переписке, когда двое участников разыгрывают одновременно десяток партий. Каждый ход записывается на отдельной открытке с указанием номера партии и номера хода. В этом случае между двумя партнерами через один и тот же почтовый канал работает как бы десяток соединений (по одному на партию). Два компьютера, связанные между собой одним физическим соединением, могут точно так же поддерживать одновременно несколько TCP-соединений. Так, например, два

промежуточных сетевых сервера могут одновременно по одной линии связи передавать друг другу в обе стороны множество TCP-пакетов от многочисленных клиентов.

Протокол IP.

Суть адресного протокола - IP (Internet Protocol) - состоит в том, что у каждого участника Всемирной сети должен быть свой уникальный адрес (IP-адрес). Без этого нельзя говорить о точной доставке TCP-пакетов на нужное рабочее место. Этот адрес выражается очень просто — четырьмя байтами, например: 195.38.46.11.

Поскольку один байт содержит до 256 различных значений, то теоретически с помощью четырех байтов можно выразить более четырех миллиардов уникальных IP-адресов (256^4 за вычетом некоторого количества адресов, используемых в качестве служебных). На практике же из-за особенностей адресации к некоторым типам локальных сетей количество возможных адресов составляет порядка двух миллиардов, но и это по современным меркам достаточно большая величина.

Протокол TCP/ IP и его основные свойства

Основой сети Интернет является стек проколов *TCP/ IP (Transmission Control Protocol/ Internet Protocol)*.

Основными преимуществами протокола TCP/ IP являются:

- *Независимость от сетевой технологии отдельной сети.* TCP/ IP не зависит от оборудования, так как он определяет только элемент передачи, который называется *дейтаграммой,* и описывает способ ее движения по сети.
- *Всеобщая связанность сетей.* Протокол позволяет любой паре компьютеров взаимодействовать друг с другом. Каждому компьютеру назначается логический адрес, а каждая передаваемая дейтаграмма содержит адреса отправителей и получателей. Промежуточные маршрутизаторы используют адрес получателя для принятия решения о маршрутизации.
- *Подтверждение.* Протокол TCP/IP обеспечивает подтверждение правильно правильности прохождения информации при обмене между отправителем и получателем.
- *Стандартные прикладные протоколы.* Протокол TCP/IP включает в свой состав поддержку основных приложений, таких как электронная почта, передача файлов, удаленный доступ и т.д.

В стеке TCP/ IP определены 4 уровня взаимодействия, каждый из которых берет на себя определенную функцию по организации надежной работы глобальной сети:

Уровень I	Прикладной уровень
Уровень II	Основной (транспортный) уровень
Уровень III	Уровень межсетевого взаимодействия
Уровень IV	Уровень сетевых интерфейсов

Уровень межсетевого взаимодействии.

Уровень межсетевого взаимодействия является стержнем всей архитектуры протокола, который реализует концепцию передачи пакетов в режиме без установления соединений, то есть дейтаграммным способом. Именно этот уровень обеспечивает возможность перемещения пакетов по сети, используя тот маршрут, который в данный момент является оптимальным. Этот уровень также называют *уровнем Интернет*, подчеркивая его основную функцию- передачу данных через составную сеть. Основным протоколом уровня межсетевого взаимодействия является протокол IP (Internet Protocol). IP – протокол проектировался для передачи пакетов в составных сетях, состоящих из большого количества локальных сетей, поэтому он хорошо работает в сетях со сложной топологией. Так как IP- протокол является дейтаграммным протоколом, то он не гарантирует доставку пакетов до узла назначения.

Основной (транспортный) уровень.

Так как на сетевом уровне не происходит установление соединения, то нет никаких гарантий, что межсетевым уровнем пакеты будут доставлены в место назначения неповрежденными. Обеспечения надежной связи между двумя конечными компьютерами осуществляет основной уровень стека TCP/IP, называемый также транспортным. На этом уровне работает протокол управления передачей TCP (Transmission Control Protocol) и протокол дейтаграмм пользователя UDP (User Datagramm Protocol). Основной задачей TCP является доставка всей информации компьютеру получателя, контроль последовательности предаваемой информации, повторная отправка не доставленных пакетов в случае сбоев работы сети. Надежность доставки информации достигается следующим образом.

На передающем компьютере TCP разбивает блок данных, поступающих с прикладного уровня, на отдельные *сегменты*, присваивает номера сегментам, добавляет заголовок и передает сегменты на уровень межсетевого взаимодействия. При этом размер сегмента должен быть таким, чтобы он полностью помещался в IP - пакет. Для каждого отправленного сегмента предающий компьютер ожидает прихода от принимающего компьютера специального сообщения – квитанции, подтверждающей тот факт, что компьютер нужный сегмент принял. Время ожидания прихода соответствующей квитанции называется *временем тайм- аута*. Переданный сегмент хранится в буфере на все время ожидания квитанции. В случае получения квитанции о правильности приема, TCP передает следующий сегмент, удаляя переданный из буфера, а в случае отсутствия квитанции о подтверждении приема, TCP повторяет передачу сегмента. Для ускорения передачи сегментов в протоколе TCP организован принцип их передачи, который называется принцип «скользящего окна». Этот принцип основывается на возможности передачи нескольких сегментов в пределах одного «окна», не дожидаясь прихода квитанции на первый отправленный сегмент. На принимающем компьютере TCP, получая от уровня межсетевого взаимодействия сегменты, собирает их в блок по номерам и

передает этот блок на верхний уровень приложений, отправляя обратно в сети квитанции о правильности принятого сегмента. Для производительности сети является очень важным установления времени тайм- аута и размера «скользящего окна». В общем случае для их выбора необходимо учитывать пропускную способность физических линий связи, отметим, однако, что в протоколе TCP предусмотрен специальный автоматический алгоритм определения этих величин.

В задачи протокола TCP входит также важнейшая задача определения к какому типу прикладных программ относятся данные, поступившие из сети. Прикладные программы с точки зрения TCP различаются специальными идентификаторами, которые называются *портами*. Назначение номеров портов осуществляется либо централизовано, если прикладные программы являются популярными и общедоступными (например, служба удаленного доступа к файлам FTP имеет порт 21, а служба WWW – порт 80), или локально – если разработчик своего приложения просто связывает с этим приложением любой доступный, произвольно выбранный номер. В дальнейшем все запросы к данному приложению от других приложений должны адресоваться с указанием назначенного ему номера порта. *Номер порта в совокупности с номером сети и номером конечного хоста однозначно определяют процессы в сети Интернет*. Этот набор идентифицирующих параметров процесса носит название *сокет*. Отметим также, что протокол TCP управляет двумя очередями: очередь пакетов, поступающих из сети и очередь пакетов, поступающих из прикладного уровня по соответствующему порту.

Протокол UDP был разработан для пользователей, не нуждающихся в услугах протокола TCP. Этот протокол, в отличие от TCP, не обеспечивает достоверность доставки пакетов и надежность от сбоев в передаче информации. К IP- пакету он добавляет только номера портов верхнего уровня. Преимущество этого протокола состоит в том, что он требует минимум установок и параметров для передачи информации и используется для наиболее простых протоколов верхнего уровня (например, для Простого протокола управления сетью - Simple Network Management Protocol, SNMP).

Прикладной уровень.

Прикладной уровень объединяет все службы пользователей сети. Прикладной уровень реализуется различными программными системами и постоянно расширяется. Наиболее известными прикладными службами являются электронная почта (E- mail), система новостей UseNet, всемирная паутина World Wide Web (WWW), передача файлов (FTP***)***, удаленный терминал и терминальные серверы (TELNET) и д.р. Указанные службы рассмотрим в следующей лекции.

Уровень сетевых интерфейсов.

В отличие от физического и канального уровня модели OSI в архитектуре стека TCP/ IP существует несколько другая интерпретация уровня сетевых

интерфейсов. Протоколы этого уровня должны обеспечить интеграцию в составную сеть локальных сетей, использующих различные технологии. Поэтому разработчики той или другой технологии должны предусмотреть возможность инкапсуляции (включения) в свои кадры IP –пакетов. Уровень сетевых интерфейсов в протоколах TCP/ IP не регламентируется, но он поддерживает все популярные стандарты физического и канального уровня: Ethernet, Token Ring, FDDI, Gigabit Ethernet, Fast Ethernet и д.р. Для глобальных сетей имеется возможность работы с протоколами SLIP и PPP. Разработаны спецификации для соединения с сетями X.25, frame relay, ATM.

Отметим, что в настоящее время каждый из разработчиков сетевых технологий канального и физического уровня стремиться обеспечить их совместимость с протоколом TCP/ IP.

Соответствие уровней стека TCP/ IP семиуровневой модели OSI

Соответствие стека TCP/IP модели OSI показано на рис 23.

Как видно из рисунка 23 протокол TCP занимает транспортный и сеансовый уровень, а на сетевом уровне используется протокол IP. Отметим, что в модели TCP/IP программные модули, соответствующие транспортному и сеансовому уровню, устанавливаются только на конечных компьютерах.

Программный модуль протокола TCP/ IP реализуется в операционной системе компьютера в виде отдельного системного модуля (драйвера). Интерфейс между прикладным уровнем и TCP представляет собой библиотеку вызовов, такую же, как, например, библиотека системных вызовов для работы с файлами. Пользователь может самостоятельно настраивать протокол TCP/ IP для каждого конкретного случая (количество пользователей сети, пропускная способность физических линий связи и т.д.).

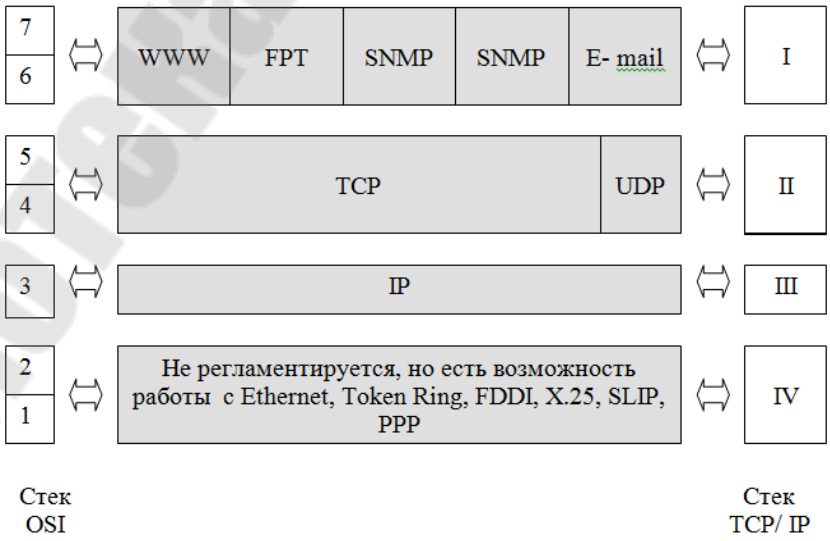


Рисунок 23 – Соответствие стека TCP/IP модели

Адресация в IP - сетях

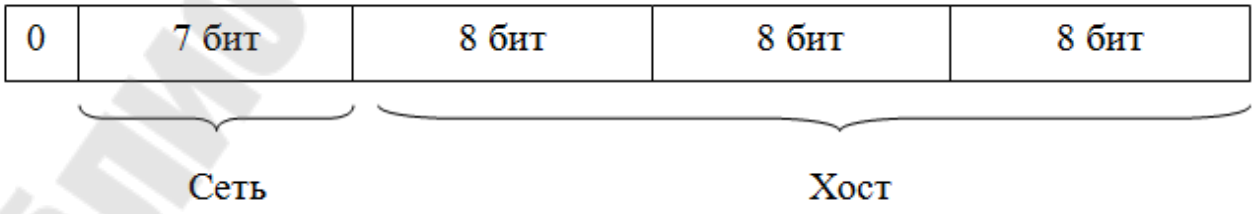
IP- адресация компьютеров в сети Интернет построена на концепции сети, состоящей из хостов. *Хост* представляет собой объект сети, который может передавать и принимать IP- пакеты, например, компьютер, рабочая станция или маршрутизатор. Хосты соединяются между собой через одну или несколько сетей. IP – адрес любого из хостов состоит *из адреса (номера) сети и адреса хоста в этой сети.*

В соответствии принятым в момент разработки IP – протокола соглашением, адрес представляется четырьмя десятичными числами, разделенными точками. Каждое из этих чисел не может превышать 255 и представляет один байт 4-байтного IP- адреса. Выделение всего лишь четырех байт для адресации всей сети Интернет связано с тем, что в то время массового распространения локальных сетей пока не предвиделось. О персональных компьютерах и рабочих станциях вообще не было речи. В результате под IP-адрес было отведено 32 бита, из которых первые 8 бит обозначали сеть, а оставшиеся 24 бита — компьютер в сети. IP – адрес назначается администратором сети во время конфигурирования компьютеров и маршрутизаторов. Номер сети может быть выбран администратором произвольным образом, или назначен по рекомендации специального подразделения Интернет – InterNIC. Обычно поставщики услуг Интернет получают диапазоны адресов у подразделений InterNIC, а затем распределяют их среди своих абонентов. Отметим, что маршрутизатор может входить сразу в несколько сетей, поэтому каждый порт маршрутизатора имеет свой IP – адрес. Таким же образом и конечный компьютер так же может входить в несколько сетей, а значит иметь несколько IP- адресов. Таким образом IP- адрес характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение.

Как уже отмечалось выше, адрес состоит из двух частей - номера сети и номера узла в сети. Для того, чтобы определить, какая часть адреса относится к номеру сети, а какая к номеру узла, в начале адреса несколько бит отводится для определения класса сети.

IP- адресация определяет пять классов сетей.

Класс А.

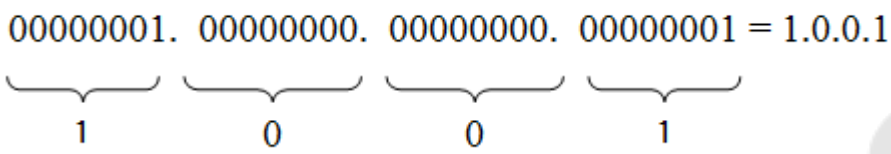


Сети класса А предназначены главным образом для использования крупными организациями, их адрес начинается с 0 в двоичной записи, или с 1 в десятичной записи, они имеют номера от 1 до 126 (если все семь бит равны «1» = 1111111= 127,

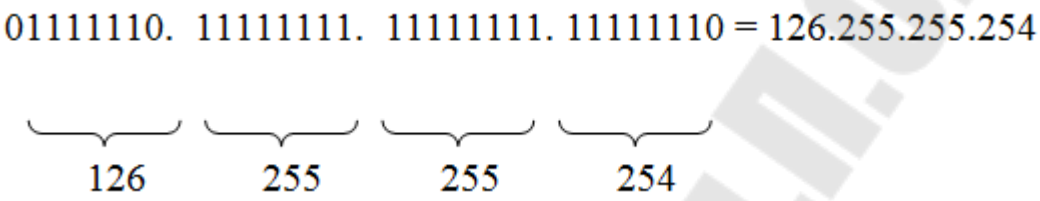
номер сети 0 не используется, а номер 127 используется для специальных целей). В сетях класса А предусмотрено большое количество узлов – $2^{24} = 16\ 777\ 216$ узлов.

Пример.

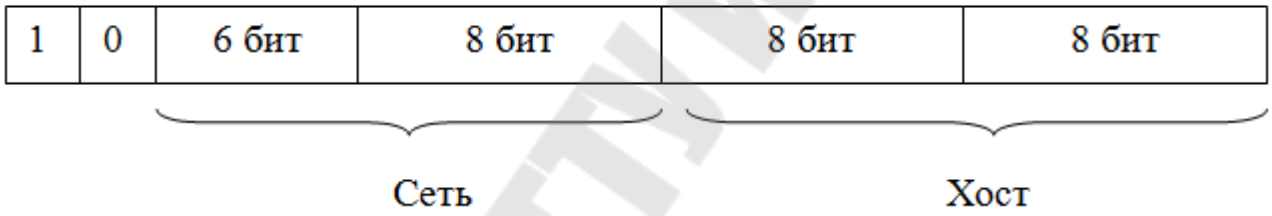
Узел имеет минимально возможный номер в сети класса А с минимально возможным номером сети



Узел имеет максимально возможный номер в сети класса А с максимально возможным номером сети



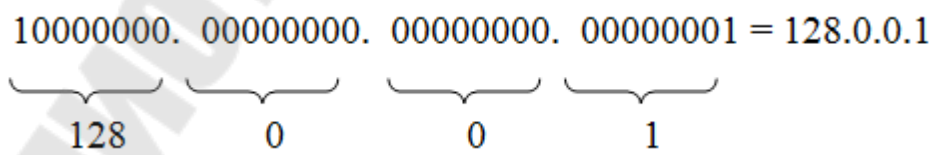
Класс В.



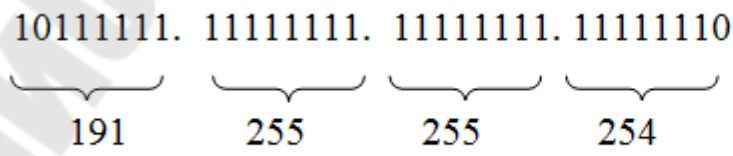
В сетях класса В выделяют 14 бит для номера сети и 16 бит для номеров хостов, их адрес начинается с 10 в двоичной записи, или со 128 в десятичной записи, они имеют номера от 128.0 до 191.255 (10000000.00000000= 128.0, 10111111.11111111= 191.255.Сети В представляют хороший компромисс между адресным пространством номера сети и номерами хостов. Сеть класса В является сетью среднего размера с максимальным числом узлов $2^{16} = 65\ 536$.

Пример.

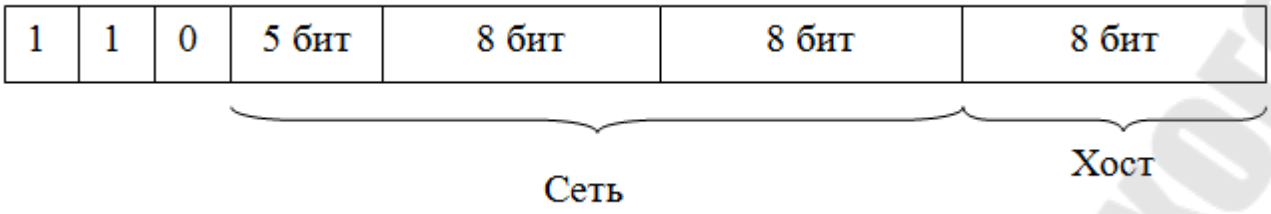
Узел имеет минимально возможный номер в сети класса В с минимально возможным номером сети



Узел имеет максимально возможный номер в сети класса В с максимально возможным номером сети



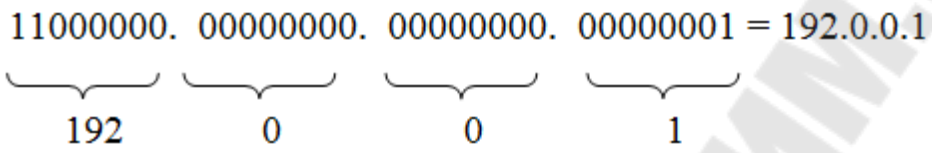
Класс С.



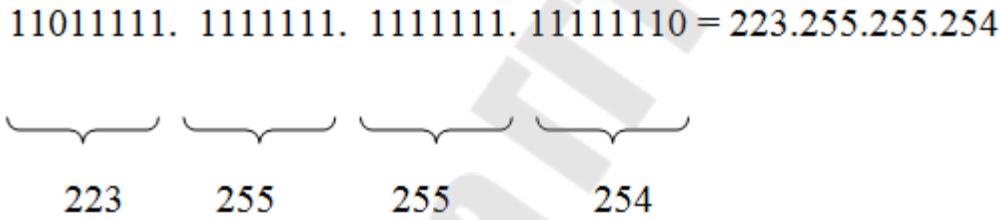
Сети класса С выделяют 22 бита для номера сети и 8 бит для номеров хостов, их адрес начинается с 110 в двоичной записи, или со 192 в десятичной записи, они имеют номера от 192.0.0 до 223.255.255 (11000000.00000000.00000000= 192.0.0, 11011111.11111111.11111111= 223.255.255. Сети класса С являются наиболее распространенными сетями, число узлов в одной сети равно $2^8 = 256$.

Пример.

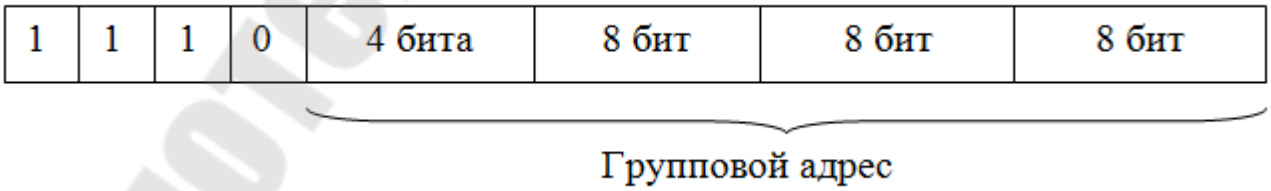
Узел имеет минимально возможный номер в сети класса С с минимально возможным номером сети



Узел имеет максимально возможный номер в сети класса С с максимально возможным номером сети



Класс D

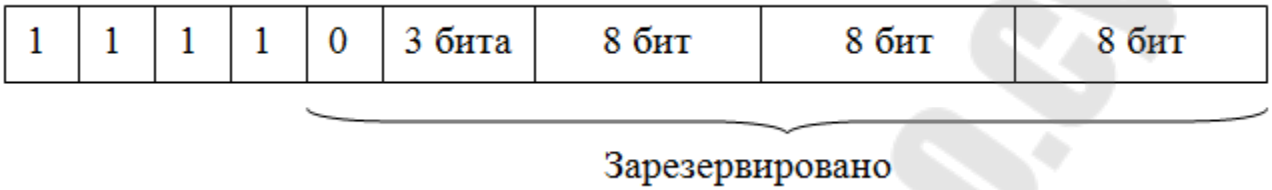


Адреса сетей класса D начинаются с 1110 в двоичной записи, или с 224 в десятичной записи, они имеют номера от 224.0.0.0 до 239.255.255.255 (11100000.00000000.00000000.00000000.=224.0.0.0, 111011111.11111111.11111111.1111111= 223.255.255.255)

Если в пакете указан адрес сети класса D, то его получат все узлы этой сети. Поэтому сети класса D называются сетями multicast – широковещательными сетями

и используются для обращения к группам узлов. Основное назначение multicast - распространение информации по схеме «один- ко- многим». Групповая адресация предназначена для экономичного распространения в Интернет или большой корпоративной сети аудио- или видеопрограмм, предназначенных сразу большой аудитории слушателей или зрителей.

Класс E



Адреса сетей класса E начинаются с 11110 в двоичной записи, или с 240 в десятичной записи, они имеют номера от 240.0.0.0 до 247.255.255.255 (11110000.00000000.00000000.00000000.=240.0.0.0, 111101111.11111111.11111111.1111111= 247.255.255.255). Сети класса E зарезервированы для будущих использований.

Некоторые IP – адреса являются выделенными и трактуются по- особому:

- Все нули – 0.0.0.0 – обозначает адрес данного узла
- Номер сети. Все нули (194.28.0.0) – данная IP- сеть
- Все нули. Номер узла (0.0.0.15) – узел в данной IP- сети
- Все единицы (255.255.255.255) – все узлы в данной IP- сети
- Номер сети. Все единицы (194.28.255.255) – все узлы в указанной IP- сети
- Число 127. единица (127.0.0.1) – «петля». Петля используется при тестировании компьютера, и данные не пересылаются по сети, а направляются на модули верхнего уровня, как будто принятые из сети. Поэтому в сетях запрещается использовать IP- адреса, начинающиеся с 127.

Использование масок в IP- адресации

Основной недостаток использования классов IP- адресов напрямую состоит в том, что если организация имеет несколько сетевых номеров, то все компьютеры вне сети имеют доступ к этим адресам и сеть организации становится прозрачной.

Для устранения указанного недостатка адресное пространство сети разбивается на более мелкие непересекающиеся пространства – подсети (subnet). С каждой из подсетей можно работать как с обычной TCP/IP – сетью.

Разбивка адресного пространства на подсети осуществляется с помощью *масок.*

Маска- это число, которое используется в паре с IP- адресом; двоичная запись маски содержит единицы в тех разрядах, которые должны в IP- адресах

интерпретироваться как номер сети. Единицы в маске должны представлять непрерывную последовательность.

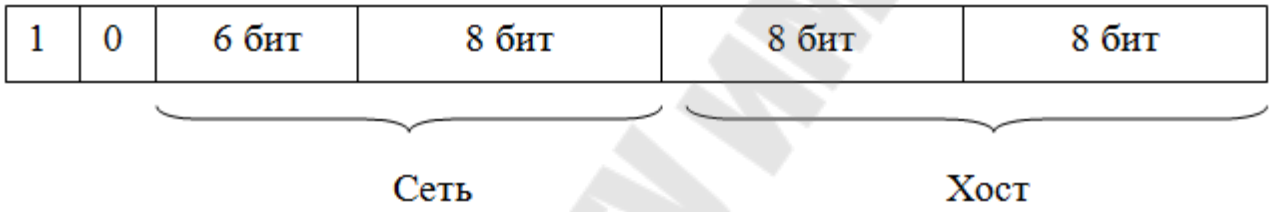
Для стандартных классов маски имеют следующие значения:

- Класс А – 11111111.00000000.00000000.00000000 (255.0.0.0)
- Класс В - 11111111.11111111.00000000.00000000 (255.255.0.0)
- Класс С - 11111111.11111111.11111111.00000000 (255.255.255.0)

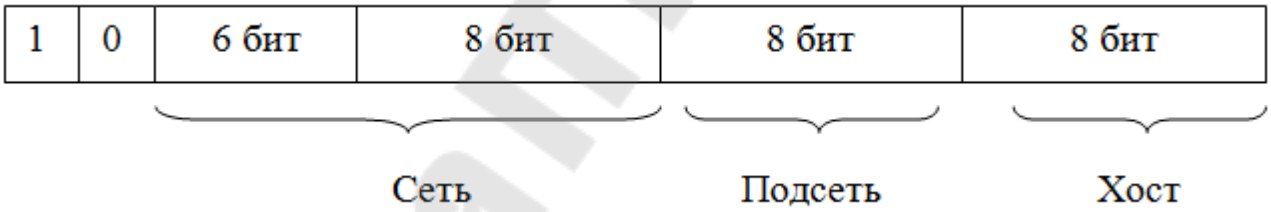
Рассмотрим, каким образом маска преобразует IP- адреса.

Пусть организация получила один IP- адрес класса В. Как известно, для сетей класса В первые два байта являются номером сети, а два остальные байта определяют номер узла. Для организации подсетей и их нумерации используются разряды байтов номеров узлов. В самом простом случае для нумерации подсетей используется первый байт номера узла.

Адрес до преобразования выглядел следующим образом:



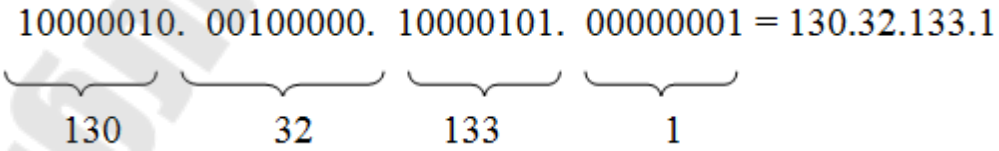
После организации подсети IP- адрес стал выглядеть:



Задавая в третьем байте номера подсети, можно разбивать сеть на отдельные подсети и присваивать номера узлов внутри подсети. В этом случае нумерация узлов внутри подсетей является локальным для организации и не видна во внешней сети. Все компьютеры вне организации видят одну большую IP- сеть и они должны поддерживать только маршруты доступа к шлюзам, соединяющим сеть организации с внешним миром.

Пример

IP- адрес сети класса В задан в виде:



а) Маска не используется. В этом случае номером сети являются первые два байта и определяют сеть 130.32.0.0, а номер узла равен 0.0.132.1

б) Используется маска:

11111111.11111111.10000000.00000000 = 255.255.128.0

В этом случае наложение маски на IP- адрес дает новое число, интерпретируемое как номер сети:

10000010. 00100000. 10000000. 00000000 = 130.32.128.0

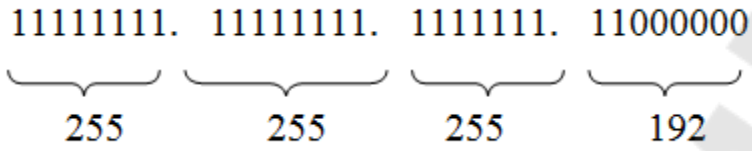
Номер узла в этой сети становится 0.0.5.1

Как видно из примера, снабжая IP-адреса маской, можно отказаться от понятий классов адресов и сделать более гибкой систему адресации сетей.

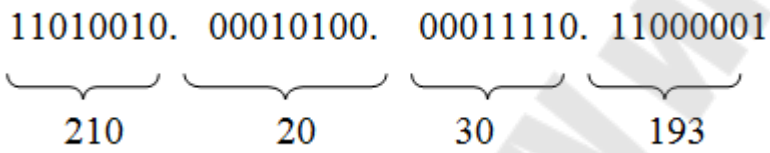
Пример

Пусть в сети работают два компьютера, имеющие два соответствующие IP-адреса: 210.20.30.193 и 210.20.30.70. Для разделения указанных компьютеров в две разные подсети используем маску 255.255.255.192

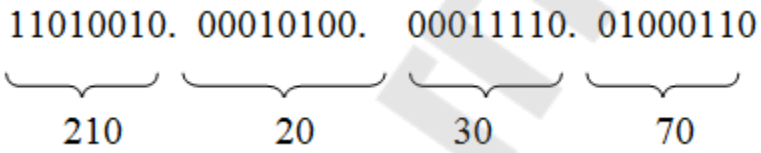
В двоичной форме маска имеет вид:



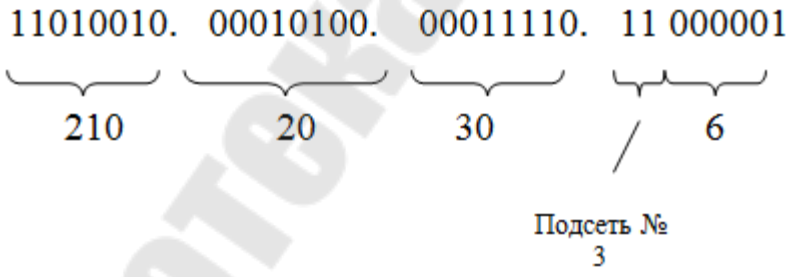
Двоичный адрес первого компьютера:



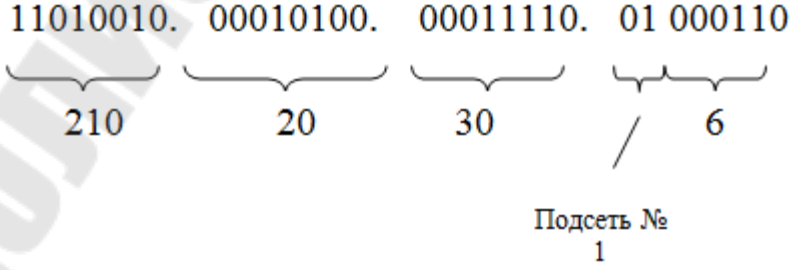
Двоичный адрес второго компьютера:



Накладывая маску на адрес первого компьютера, получим его новый адрес:



Накладывая маску на адрес второго компьютера, получим его новый адрес:



Таким образом, сеть с помощью маски разбилась на две подсети, номер второго компьютера в подсети стал равным шести.

Следует отметить, что в настоящее время наблюдается дефицит IP- адресов, выделяемых организацией InterNIC. Очень трудно получить адрес класса В и практически невозможно стать обладателем адреса класса А. Если же IP- сеть создана для работы в автономном режиме, без связи с Интернет, то администратор сети сам произвольно назначает номер. Но даже в этой ситуации в стандартах Интернет определены несколько диапазонов адресов, не рекомендуемых для использования в локальных сетях. Эти адреса не обрабатываются маршрутизаторами Интернет ни при каких условиях. Для сетей класса А – это сеть 10.0.0.0, в классе В- это диапазон из 16 номеров сетей 172.16.0.0 – 172.31.0.0, в классе С – это диапазон из 255 сетей – 192.168.0.0 – 192.168.255.0.

Для разрешения проблемы дефицита адресов осуществляется переход на новую версию IP- протокола- протокол IPv6, в котором резко расширяется адресное пространство за счет 16- байтных адресов.

Протокол IPv6, как развитие транспортных средств IP- протокола

Указанный протокол решает принципиальную проблему нехватки IP-адресов посредством использования 128- разрядных адресов вместо 32 – разрядных адресов, благодаря чему адресное пространство расширяется в 296 раз. Результатом этого будет то, что любой житель Земли может получить в сове распоряжение несколько IP- адресов, новое количество адресов позволит подключить к сети свыше 1 квадрильона компьютеров в 1 триллионе сетей.

Адреса в IPv6 – протоколе разделяются на три типа: *обычные, групповые и нечеткие.*

Пакет с обычным адресом передается конкретному адресату, в то время как пакет с групповым адресом доставляется всем членам группы. Пакет с нечетким адресом доставляется только ближайшему члену данной группы.

В IPv6 128 разрядные адреса записываются в виде восьми 16- разрядных целых чисел, разделенных двоеточие. Каждое число представлено шестнадцатеричными цифрами, разделенными двоеточиями. Другими словами, необходимо вводить 32 шестнадцатеричные цифры для задания IP- адреса. IPv6 – адрес может выглядеть так: 501A:0000:0000:0000:00FC:ABCD:3F1F:3D5A.

Переход от традиционных IP- адресов к IPv6 – адресам займет ни один год и старая адресация будет постепенно замещаться новыми программными продуктами и оборудованием, использующим IPv6- протокол.

Среди других новых свойств IPv6 – протокола можно отметить также более рациональную структуру формата заголовка пакета, увеличение производительности маршрутизаторов, работающих с этим протоколом, возможность маркировки потока данных, если их необходимо обрабатывать особым образом, аутентификацию дейтаграмм и д.р.

Протокол ICMP

Межсетевой протокол управляющих сообщений (ICMP - Internet Control Message Protocol) разработан для того, чтобы маршрутизаторы в Интернете сообщали об ошибках или предоставляли информацию о нестандартных условиях работы сети. Он является необходимой частью протокола IP. И обеспечивает обратную связь, оповещение отправителя данных о проблемах, возникающих в коммуникационном оборудовании.

Протокол ICMP - это механизм сообщения об ошибках. Он обеспечивает маршрутизаторам, обнаруживающим ошибки, способ сообщения об ошибке первоначальному источнику. Хотя спецификация протокола определяет допустимые способы использования ICMP и предлагает варианты возможных действий в ответ на ошибки, ICMP не специфицирует полностью действия, которые необходимо предпринять в ответ на все возможные ошибки. Таким образом, ICMP только сообщает о возникших ошибках первоначальному источнику; источник сам должен связать ошибки с конкретными прикладными программами и предпринять действия по исправлению ошибок.

Протокол ICMP выполняет следующие основные функции:

- обмен тестовыми сообщениями для выяснения наличия и активности узлов сети;
- анализ достижимости узлов и сброс пакетов, направленных к недостижимым узлам;
- изменение маршрутов (Redirect);
- уничтожение пакетов с истекшим временем жизни (Time-To-Live);
- синхронизация времени в узлах сети;
- управление трафиком (регулирование частоты отправки пакетов).

С точки зрения уровневых протоколов, ICMP является частью сетевого уровня. Но по отношению к IP ICMP протокол более высокого уровня, так как он пользуется услугами IP для доставки своих сообщений. Другими словами, каждое сообщение ICMP передается по сети внутри 1P-дейтаграммы.

ICMP-сообщения бывают двух видов: сообщение-запрос и сообщение об ошибке. Сообщение об ошибке тесно связано с породившей его дейтаграммой и всегда содержит заголовок этой IP-дейтаграммы и первые 64 бит ее данных. Это необходимо для того, чтобы узел-источник смог более точно проанализировать причину ошибки, так как все прикладные протоколы стека TCP/IP содержат наиболее важную информацию для анализа в первых 8 байт своего сообщения. Сообщения-запросы передают информацию об определенной сети и об определенном компьютере или их используют для диагностических целей.

IP-пакеты с сообщениями ICMP передаются по сети «на общих основаниях», без приоритетов, поэтому они тоже могут теряться. В загруженной сети они могут

вызвать дополнительную загрузку маршрутизаторов, когда потеря сообщения об ошибке приводит к порождению нового сообщения и т. д., пока канал связи не исчерпает своей пропускной способности. Для того чтобы предотвратить подобные ситуации, в спецификации четко определены правила, руководствуясь которыми компьютер может решить, передавать его ICMP- сообщение или нет.

Правило 1: потеря пакета с ICMP-сообщением никогда не генерирует нового ICMP-сообщения.

Правило 2: сообщения об ошибке никогда не генерируются в ответ на IP-дейтаграммы с широковещательными или групповыми адресами, чтобы не вызвать полную перегрузку сети - широковещательный шторм (broadcast storm).

Правило 3: при повреждении фрагментированной дейтаграммы, 1CMP-сообщение отправляют только после первого поврежденного фрагмента (так как источник все равно повторит передачу всей дейтаграммы целиком).

Доставка ICMP-пакетов требует двух уровней инкапсуляции. ICMP-пакеты инкапсулируются внутрь IP-дейтаграммы, которая сама передается по каждой физической сети в поле данных кадра (рис. 24).

Рисунок 24 – Два уровня инкапсуляции сообщения ICMP

Несмотря на то, что сообщения ICMP инкапсулируются и посылаются, используя IP, ICMP не считают протоколом более высокого уровня - он является необходимой частью IP. Протокол IP необходим для транспортировки сообщений ICMP, так как им, чтобы достичь своего конечного назначения, надо пересечь несколько физических сетей. Поэтому, они не могут быть доставлены только с помощью физической передачи.

Формат ICMP-пакетов. Хотя каждое сообщение ICMP имеет свой собственный формат, все они начинаются с трех одинаковых полей: 8-битового целочисленного поля «Тип», идентифицирующего сообщение, 8-битового поля «Код», обеспечивающего более точную информацию о типе сообщения, и 16битового поля «Контрольная сумма» (рис. 25). Помимо того, сообщения ICMP, сообщающие об ошибках, всегда включают заголовок и первые 64 бит данных дейтаграммы, вызвавшей ошибку. Это необходимо для того, чтобы узел-от-правитель смог более точно проанализировать причину ошибки, так как все протоколы прикладного уровня стека TCP/IP содержат наиболее важную информацию для анализа в первых 64 бит своих сообщений.

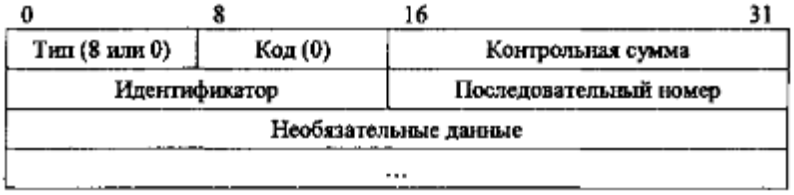


Рисунок 25 – Формат пакета ICMP

Сетевые программы распознают ICMP-сообщения по двум признакам: значению поля «Тип» и значению поля «Код». Контрольная сумма вычисляется не только для ICMP-заголовка, но и для всего сообщения.

Таблица 4 – Типы сообщений ICMP

Тип сообщения ICMP	Описание
0	Ответ на эхо (Echo Reply)
3	Узел назначения недостижим (Destination Unreachable)
4	Подавление источника (Source Quench)
5	Перенаправление маршрута (Redirect)
8	Запрос эха (Echo Request)
9	Информация о маршрутизаторах (Router Advertisement)
10	Регистрация маршрутизатора (Router Solicitation)
И	Лимит времени для дейтаграммы превышен (Time Exceeded for a Data- gramm)
12	Проблема с параметром пакета (Parameter Problem on a Datagramm)
13	Запрос метки времени (Timestamp Request)
14	Ответ для метки времени (Timestamp Reply)
15	Запрос информации (не действует)
16	Ответ на запрос информации (не действует)
17	Запрос маски адреса (Address Mask Request)
18	Ответ на запрос маски адреса (Address Mask Reply)

Типы сообщений ICMP представлены в табл. 4. Рассмотрим каждое из этих сообщений и его формат подробнее.

Тестирование достижимости места назначения. Протоколы TCP/IP обеспечивают средства, помогающие сетевым администраторам или пользователям идентифицировать проблемы в сети. Пользователь в качестве одного из широко используемых средств отладки применяют команду, которая вызывает сообщения ICMP запроса эха и ответа эха. Компьютер или маршрутизатор посылает сообщение запроса эха указанному месту назначения. Любая машина, получившая запрос эха, генерирует ответ на эхо и возвращает его первоначальному отправителю. Этот запрос содержит необязательную область данных; ответ содержит копию данных, посланных в запросе. Запрос эха и связанный с ним ответ можно использовать для проверки достижимости назначения и его способности отвечать на запросы. Так как и запрос эха, и ответ на него передаются в IP-дейтаграммах, успешный прием ответа свидетельству-ет о работоспособности основных частей транспортной системы. Во-первых, программное обеспечение IP на машине источника выполнило

маршрутизацию дейтаграммы. Во-вторых, промежуточные маршрутизаторы между источником и получателем работоспособны и корректно маршрутизируют дейтаграммы. В-третьих, машина получателя работает (по крайней мере, она обрабатывает прерывания) и программное обеспечение, как IP, так и ICMP, выполняет свои функции. И, наконец, таблицы маршрутов в маршрутизаторах на всем обратном пути корректны.

Во многих системах команда, которую пользователи вызывают для посылки запроса эха ICMP, называется ping. Усложненные версии этой программы посылают серии запросов эха ICMP, принимают ответы и выдают статистику о потерях дейтаграмм. Они позволяют пользователю указывать длину посылаемых данных и интервалы времени между запросами. Менее сложные версии просто посылают запрос эха ICMP и ждут ответа.

Формат сообщения запроса эха и ответа эха. Средства для тестирования достижимости узлов сети представляют собой очень простой эхо-протокол, включающий обмен двумя типами сообщений: эхо-запрос и эхо-ответ. Компьютер или маршрутизатор посылают по интерсети эхо-запрос, в котором указывают IP-адрес узла, достижимость которого нужно проверить. Узел, получающий эхо-запрос, формирует и отправляет эхо-ответ и возвращает сообщение узлу - отправителю запроса. В запросе могут содержаться некоторые данные, которые должны быть возвращены в ответе.

Рис. 25 иллюстрирует формат сообщений запроса эха и ответа на запрос эха. Поле «Необязательные данные» имеет переменную длину и содержит данные, которые надо вернуть отправителю. Ответ на эхо всегда возвращает те же самые данные, что были получены им в запросе. Поля «Идентификатор» и «Последовательный номер» отправитель использует для проверки соответствия ответов запросам. Значение поля «Тип» определяет, является ли сообщение запросом (8) или ответом (0).

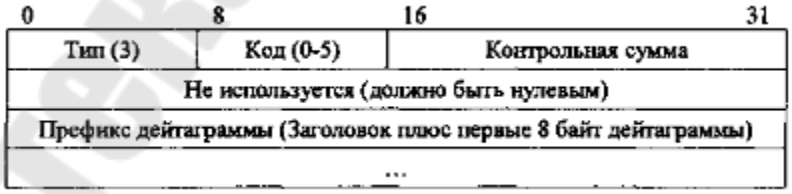


Рисунок 26 – Формат сообщения о недостижимости назначения

Сообщения о недостижимости назначения. Когда маршрутизатор не может доставить IP-дейтаграмму, он посылает сообщение «назначение недостижимо» первоначальному отправителю, используя формат, приведенный на рис. 26. Поле «Код» в сообщении о недостижимости назначения содержит целое число, которое описывает причину. Возможные значения представлены в табл. 5.

Таблица 5 – Коды сообщений о недостижимости

Код сообщения	Пояснения
0	Сеть недостижима
1	Компьютер недостижим
2	Протокол недостижим
3	Порт недостижим
4	Необходима фрагментация
5	Ошибка при маршрутизации источника
6	Сеть назначения неизвестна
7	Компьютер назначения неизвестен
8	Компьютер источника изолирован
9	Взаимодействие с сетью назначения административно запрещено
10	То же с компьютером назначения
И	Сеть недостижима из-за класса обслуживания
12	Компьютер недостижим из-за класса обслуживания

Хотя протокол ЕР является механизмом ненадежной доставки, дейтаграммы не уничтожаются просто так. Всякий раз, когда ошибка мешает маршрутизатору произвести маршрутизацию или доставку дейтаграммы, маршрутизатор посылает сообщение о недостижимости назначения его источнику, а затем уничтожает дейтаграмму. Ошибки недостижимости сети обычно являются следствием ошибок маршрутизации; ошибки недостижимости компьютера - следствие ошибок при доставке.

Назначения могут быть недостижимыми из-за того, что оборудование было временно неработоспособно, отправитель указал несуществующий адрес назначения или (в редких случаях) у маршрутизатора не указано маршрута к сети назначения. Необходимо отметить, что не все подобные ошибки можно обнаружить.

Если дейтаграмма содержит опцию маршрутизации источника с некорректным маршрутом, то это может привести к появлению сообщения об ошибке маршрутизации источника. Если шлюзу нужно фрагментировать дейтаграмму, но установлен бит «не фрагментировать», то шлюз посылает сообщение «требуется фрагментация» обратно источнику.

Управление потоком дейтаграмм и переполнение сети. Так как IP- протокол не устанавливает соединения, то маршрутизаторы не могут резерви-ровать память или коммуникационные ресурсы до получения дейтаграмм. В результате, трафик может вызвать перегрузку маршрутизаторов, ситуацию, называемую переполнением сети (congestion). Переполнение сети происходит по двум совершенно разным причинам. Во-первых, высокоскоростной компьютер может генерировать трафик быстрее, чем сеть может передавать его. Например, представим суперкомпьютер, генерирующий межсетевой трафик. Дейтаграммам, посылаемым им, может потребоваться передача, в конечном счете, по медленной глобальной сети (WAN), хотя сам суперкомпьютер может быть присоединен к высокоскоростной LAN. Переполнение будет возникать в маршрутизаторе, присоединенном к глобальной сети, так как дейтаграммы будут

прибывать быстрее, чем их можно послать. Во-вторых, если большому числу компьютеров одновременно нужно посылать дейтаграммы через один маршрутизатор, этот маршрутизатор может оказаться переполненным, хотя ни один источник в отдельности не вызывает эту проблему.

Когда дейтаграммы прибывают на шлюз или маршрутизатор быстрее, чем он успевает их обрабатывать, он временно ставит их в очередь в своей памяти. Если эти дейтаграммы создают небольшую пиковую нагрузку при передаче дейтаграмм, то такая буферизация решает проблему. Если же трафик продолжает поступать, то, в конечном счете, маршрутизатор или шлюз займет всю память под очередь и вынужден будет удалять новые прибывающие дейтаграммы. Тогда машина для выхода из состояния переполнения использует сообщения о подавлении источника.

Сообщение о подавлении источника требует от источника уменьшить скорость передачи дейтаграмм. Обычно переполненные маршрутизаторы посылают по одному сообщению о подавлении источника на каждую удаляемую дейтаграмму или используют более сложные технологии выхода из переполнения. Формат подавления источника представлен на рис. 27. Помимо обычных полей ICMP «Тип», «Код» и «Контрольная сумма» и неиспользуемого 32-битового поля, сообщения о подавлении источника имеют поле, содержащее префикс дейтаграммы. Как и в других сообщениях об ошибках ICMP поле префикса дейтаграммы содержит префикс дейтаграммы, вызвавшей этот запрос подавления источника.

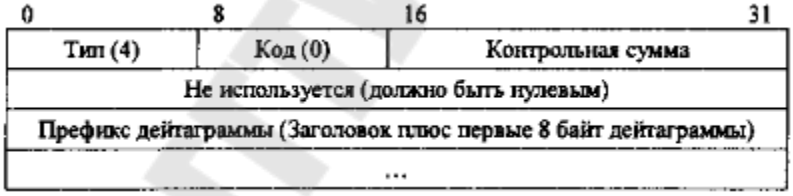


Рисунок 26 – Формат сообщения о подавлении источника ICMP

Сообщения ICMP, вызывающего эффект, обратный подавлению источника, не существует. Вместо этого, компьютер, принявший сообщения о подавлении источника от некоторой машины, снижает скорость, с которой он посылает ей дейтаграммы. Это происходит до тех пор, пока к нему не перестанут приходить сообщения о подавлении источника. Затем он постепенно увеличивает скорость пока снова не получит сообщения о подавлении источника.

Перенаправление маршрута. Маршрутные таблицы у компьютеров обычно статические, так как их конфигурирует администратор сети, а у маршрутизаторов - динамические, формируемые автоматически с помощью протоколов обмена маршрутной информации. Поэтому с течением времени при изменении топологии сети маршрутные таблицы компьютеров могут устаревать.

При изменении топологии сети таблицы маршрутизации в маршрутизаторе или компьютере могут стать некорректными. Изменение может быть временным (например, нужно заменить неисправное оборудование) или постоянным (например,

когда в межсетевое взаимодействие включается новая сеть). Маршрутизаторы периодически обмениваются информацией о маршрутизации, чтобы отслеживать изменения в сети и своевременно менять маршруты. Для корректировки поведения компьютеров маршрутизатор может использовать сообщение протокола ICMP, называемое «перенаправлением» (Redirect), запрашивающее изменение маршрута в таблице маршрутизации компьютера.

Механизм перенаправления протокола ICMP позволяет компьютерам содержать в конфигурационном файле только IP-адреса его локальных маршрутизаторов. С помощью сообщений о перенаправлении маршрутизаторы будут сообщать компьютеру всю необходимую ему информацию о том, какому маршрутизатору следует отправлять пакеты для той или иной сети назначения, т. е. маршрутизаторы передадут компьютеру нужную ему часть их таблиц маршрутизации.

Преимуществом схемы перенаправления ICMP является ее простота: она позволяет компьютеру знать вначале адрес только одного маршрутизатора в локальной сети. Этот начальный маршрутизатор возвращает сообщение ICMP о перенаправлении всякий раз, когда компьютер посылает дейтаграмму, для которой существует лучший маршрут. Таблица маршрутизации компьютера останется маленькой, но содержит оптимальные маршруты для всех используемых назначений.

Сообщения о перенаправлении, тем не менее, не решают проблему распространения информации о маршрутах полностью, так как они предназначены только для взаимодействия между маршрутизатором и компьютером в одной физической сети. Каждое сообщение о перенаправлении содержит 32-битовое поле «IP-адрес маршрутизатора» и поле «Префикс дейтаграммы», как это показано на рис. 27.

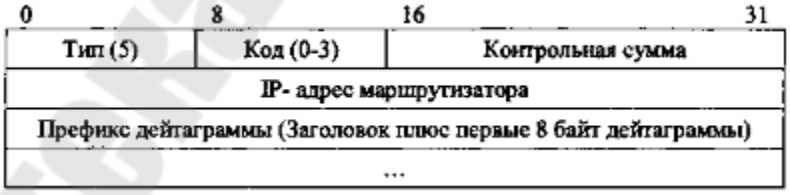


Рисунок 27 – Формат сообщения о перенаправлении ICMP

Поле «Межсетевой адрес маршрутизатора» содержит IP-адрес маршрутизатора, который должен использовать компьютер при отправлении дейтаграммы к назначению, указанному в заголовке дейтаграммы. Поле «Префикс дейтаграммы» содержит заголовок IP и следующие 8 байт дейтаграммы, которая привела к появлению этого сообщения. Поэтому компьютер, принимающий сообщение о перенаправлении ICMP, должен выделить адрес назначения дейтаграммы из префикса дейтаграммы. Поле «Код» в сообщении о перенаправлении ICMP более конкретно указывает, как интерпретировать адрес

назначения, при этом значения имеют следующий смысл: 0 - перенаправление дейтаграмм для этой сети (устарело), 1 - перенаправление дейтаграмм для этого компьютера, 2 - перенаправление дейтаграмм для этого типа сервиса и сети, 3 - перенаправление дейтаграмм для этого типа сервиса и компьютера. Напомним, что каждый заголовок IP указывает тип сервиса, используемого при маршрутизации. Как правило, маршрутизаторы посылают запросы переназначения ICMP только на компьютеры, а не на другие маршрутизаторы.

Изменение маршрута является одной из наиболее интересных функций протокола ICMP - по существу, это один из механизмов автоматической оптимизации доставки пакетов и адаптации сетей TCP/IP к изменениям топологии.

Запросы «Информация о маршрутизаторах» (типы 9 и 10).

Информация о маршрутизации находится в местных конфигурационных файлах и загружается оттуда при запуске компьютера. Чтобы таблица маршрутизации не содержала устаревших данных она обновляется динамически. ICMP- протокол реализует один из способов ее обновления.

Существует 2 типа сообщений маршрутизаторов:

- 9 - информация о маршрутизации;
- 10-регистрация маршрутизатора.

Всякий раз, когда компьютер запускают, он генерирует сообщения о регистрации. В ответ маршрутизаторы, находящиеся в той же локальной сети, посылают сообщения с информацией о маршрутизации, позволяющие правильно сконфигурировать маршрутную таблицу.

Формат сообщения «Информация о маршрутизации» (тип 9) описан в RFC 1256 (рис. 28).

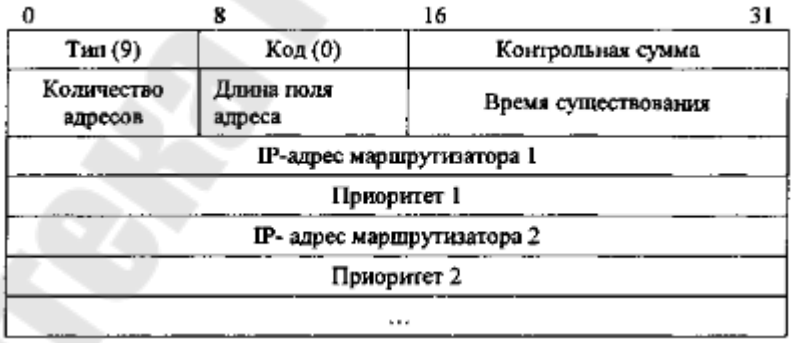


Рисунок 28 – Формат сообщения «Информация о маршрутизации»

В одном ICMP-сообщении может содержаться описание нескольких адресов, количество которых указано в поле «Количество адресов». Поле «Размер адреса» задает длину адреса в 32-битовых словах. В настоящее время «Длина поля адреса» всегда равна 2.

Поле «Время существования» задает интервал времени, в течение которого информация еще не устарела. Как правило, это 1800 с.

Поле «Приоритет» указывает, какой из адресов следует использовать первым и более интенсивно. Как правило, чем больше значение поля, тем выше приоритет. Маршрутизаторы передают информационные сообщения широковещательно через случайные интервалы времени. Обычно через 450...600 с. Поле «Время существования» можно использовать для уведомления, что данный маршрутизатор выключается. При этом содержимое данного поля устанавливается равным 0.

Формат сообщения «Регистрация» (тип 10) представлен на рис. 29.

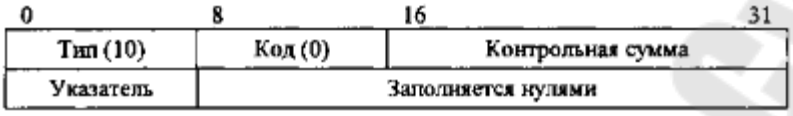


Рисунок 29 – Формат сообщения «Регистрация»

Запрос «Регистрация» передается 3 раза с интервалом 3 с при запуске маршрутизатора и продолжает (при необходимости) передаваться, пока маршрутизатор не получит информационного сообщения с нужной маршрутной информацией.

Обнаружение циклических или слишком длинных путей. Как было отмечено выше для защиты Интернета от перегрузок каждая дейтаграмма имеет счетчик времени жизни TTL (Time-To-Live). Маршрутизатор декрементирует счетчик времени жизни всякий раз, когда он обрабатывает дейтаграмму, и удаляет ее, когда счетчик становится нулевым.

Независимо от того, удалил ли маршрутизатор дейтаграмму из-за обнуления счетчика времени жизни или из-за превышения времени ожидания фрагментов дейтаграммы, он посылает сообщение ICMP «Лимит времени для дейтаграммы превышен» источнику дейтаграммы определенного формата (рис. 30).

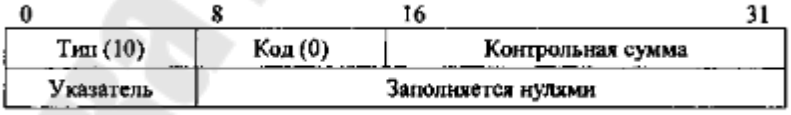


Рисунок 30 – Формат сообщения «Лимит времени для дейтаграмм превышен»

Поле «Код» объясняет причину сообщения: 0 - превышено значение счетчика времени жизни; 1 - превышено время ожидания фрагмента при сборке.

Сообщения о других ситуациях. Когда маршрутизатор или компьютер сталкивается с проблемой, не укладывающейся в рамки описанных сообщений об ошибках ICMP (например, некорректный заголовок дейтаграммы), связанной с дейтаграммой, он посылает сообщение «Проблема с параметром пакета» первоначальному отправителю. Такую ситуацию может вызвать некорректность аргументов опции. Сообщение, формат которого показан на рис. 31, посылается только в том случае, если дейтаграмма должна быть удалена из- за этой ошибки. Для уточнения места ошибки в дейтаграмме отправитель использует поле

«Указатель» в заголовке сообщения для идентификации октета в дейтаграмме, содержащего ошибку.

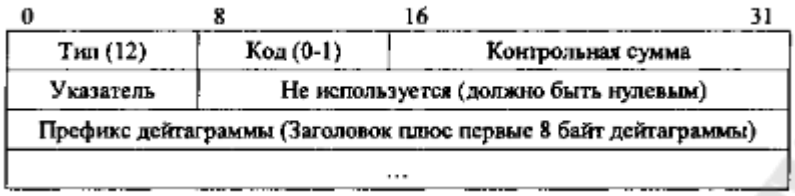


Рисунок 31 – Формат сообщения «Проблемы с параметром пакета»

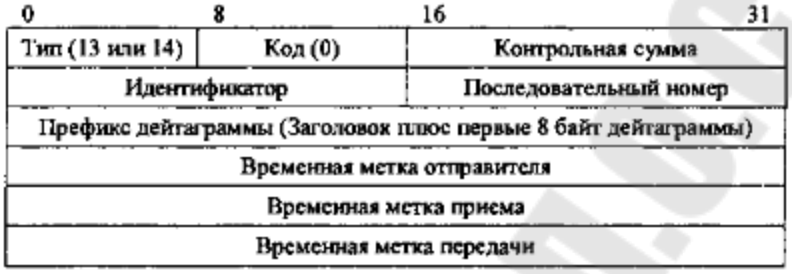


Рисунок 32 – Формат сообщения «Запрос метки времени»
и «Ответ для метки времени»

Синхронизация часов и оценка времени передачи. Стек протоколов TCP/IP включает несколько протоколов, которые могут использоваться для синхронизации часов. В сети для этого используется несколько технологий. Одна из простейших технологий реализуется сообщениями ICMP для получения значения времени от другой машины. Запрашивающая машина посылает сообщение ICMP «Запрос метки времени» другой машине, ожидая, что вторая машина вернет ей текущее значение времени. Принимающая машина возвращает «Ответ для метки времени» машине, выдавшей запрос. Рис. 32 иллюстрирует формат сообщений запроса и ответа временной метки. Поле «Тип» идентифицирует сообщение как запрос (13) или ответ (14); поля «Идентификатор» и «Последовательный номер» используют источник для связи между ответами и запросами. Оставшиеся поля специфицируют времена, указанные в миллисекундах после полуночи, по Гринвичу. Поле «Временная метка отправителя» заполняет первоначальный отправитель перед передачей пакета, поле «Временная метка приема» заполняется сразу после приема запроса, а поле «Временная метка передачи» - непосредственно перед отправкой ответа.

Компьютеры используют эти три поля временных меток для определения ожидаемого времени передачи между ними и синхронизации своих часов. Так как ответ включает поле «Временная метка отправителя», компьютер может вычислить общее время, требуемое для передачи запроса к назначению, формирования ответа на него и возвращения ответа. Так как ответ содержит как время прихода запроса на удаленную машину, так и время выдачи ответа, компьютер может вычислить время передачи по сети, а на его основе - разницу между своими и удаленными часами. На практике бывает трудно точно оценить время передачи по сети, так как ГР является технологией с негарантированной доставкой, дейтаграммы могут бггь потеряны,

задержаны или доставлены не по порядку, что ограничивает полезность сообщений ICMP о временных метках.

Сообщения запроса и ответа информации. Сообщения ICMP запроса информации и ответа информации (тип 15 и 16) в настоящее время устарели и их использовать не рекомендуется. Они предназначались для обнаружения компьютерами своих IP-адресов при загрузке. Сейчас для определения адреса используют протоколы RARP и BOOTP.

DHCP

DHCP (англ. Dynamic Host Configuration Protocol — протокол динамической конфигурации узла) — это сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP. Данный протокол работает по модели «клиент-сервер». Для автоматической конфигурации компьютер-клиент на этапе конфигурации сетевого устройства обращается к так называемому серверу DHCP, и получает от него нужные параметры. Сетевой администратор может задать диапазон адресов, распределяемых сервером среди компьютеров. Это позволяет избежать ручной настройки компьютеров сети и уменьшает количество ошибок. Протокол DHCP используется в большинстве сетей TCP/IP.

DHCP является расширением протокола BOOTP, использовавшегося ранее для обеспечения бездисковых рабочих станций IP-адресами при их загрузке. DHCP сохраняет обратную совместимость с BOOTP.

Протокол DHCP предоставляет три способа распределения IP-адресов:

Ручное распределение. При этом способе сетевой администратор сопоставляет аппаратному адресу (для Ethernet сетей это MAC-адрес) каждого клиентского компьютера определённый IP-адрес. Фактически, данный способ распределения адресов отличается от ручной настройки каждого компьютера лишь тем, что сведения об адресах хранятся централизованно (на сервере DHCP), и потому их проще изменять при необходимости.

Автоматическое распределение. При данном способе каждому компьютеру на постоянное использование выделяется произвольный свободный IP-адрес из определённого администратором диапазона.

Динамическое распределение. Этот способ аналогичен автоматическому распределению, за исключением того, что адрес выдаётся компьютеру не на постоянное пользование, а на определённый срок. Это называется арендой адреса. По истечении срока аренды IP-адрес вновь считается свободным, и клиент обязан запросить новый (он, впрочем, может оказаться тем же самым). Кроме того, клиент сам может отказаться от полученного адреса.

Некоторые реализации службы DHCP способны автоматически обновлять записи DNS, соответствующие клиентским компьютерам, при выделении им новых адресов. Это производится при помощи протокола обновления DNS.

Помимо IP-адреса, DHCP также может сообщать клиенту дополнительные параметры, необходимые для нормальной работы в сети. Эти параметры называются опциями DHCP.

Некоторыми из наиболее часто используемых опций являются:

- IP-адрес маршрутизатора по умолчанию;
- маска подсети;
- адреса серверов DNS;
- имя домена DNS.

Некоторые поставщики программного обеспечения могут определять собственные, дополнительные опции DHCP.

Все сообщения протокола DHCP разбиваются на поля, каждое из которых содержит определённую информацию(рис.33). Все поля, кроме последнего (поля опций DHCP), имеют фиксированную длину.

Поле	Описание
op	Тип сообщения (1 = BOOTREQUEST, 2 = BOOTREPLY)
htype	Тип адреса оборудования
hlen	Длина адреса оборудования
hops	Используется ретранслирующим агентом
xid	Идентификатор транзакции между сервером и клиентом
secs	Время с момента выдачи DHCPREQUEST или начала обновления конфигурации
flags	Флаги (первый бит маркирует широковещательные сообщения)
ciaddr	IP-адрес клиента
yiaddr	<Ваш> (клиентский) IP-адрес
siaddr	IP-адрес следующего сервера, участвующего в загрузке
giaddr	IP-адрес ретранслирующего агента
chaddr	<Аппаратный> адрес клиента
sname	Хост-имя сервера (опция)
file	Имя загрузочного файла
options	Поле дополнительных параметров

Рисунок 33 – Поля DHCP

Протокол DHCP является клиент-серверным, то есть в его работе участвуют клиент DHCP и сервер DHCP. Передача данных производится при помощи протокола UDP, при этом сервер принимает сообщения от клиентов на порт 67 и отправляет сообщения клиентам на порт 68.

Работа протокола DHCP базируется на классической схеме клиент-сервер. В роли клиентов выступают компьютеры сети, стремящиеся получить IP-адреса в так называемую аренду (lease), а DHCP-серверы выполняют функции диспетчеров, которые выдают адреса, контролируют их использование и сообщают клиентам требуемые параметры конфигурации. Сервер поддерживает пул свободных адресов и, кроме того, ведет собственную регистрационную базу данных. Взаимодействие

DHCP-серверов со станциями-клиентами осуществляется путем обмена сообщениями.

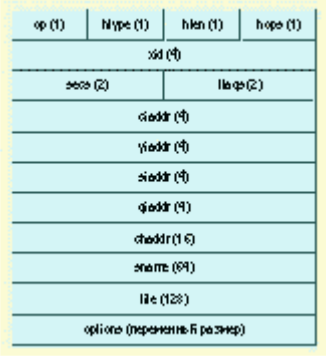


Рисунок 34 – Формат сообщения DHCP (в скобках - размер поля в байтах)

DHCP разрабатывался как непосредственное расширение BOOTP и именно в таком качестве воспринимается BOOTP-клиентами. Этому обстоятельству в первую очередь способствует формат сообщений DHCP, во многом совпадающий с форматом, который применяется протоколом-предшественником и определен в документе RFC 951 (рис. 34).

Сравнивая протоколы BOOTP и DHCP, нельзя не отметить появления в DHCP новых услуг. Во-первых, вэтом протоколе предусмотрен механизм автоматической выдачи IP-адресов во временное пользование с возможностью их последующего присвоения новым клиентам. Во-вторых, клиент может получить от сервера все параметры конфигурации, которые ему необходимы для успешного функционирования в IP-сети.

Указанные отличия потребовали частичного расширения формата сообщений. Так, в нем появилось отдельное поле идентификатора клиента, сделана более прозрачной интерпретация адреса сервера (поле siaddr), переменный размер получило поле options, используемое, в частности, для передачи параметров конфигурации (его длина обычно находится в диапазоне 312-576 байт, хотя возможно и дополнительное расширение этого поля за счет полей sname и file).

В роли транспортного протокола для обмена DHCP-сообщениями выступает UDP. При отправке сообщения с клиента на сервер используется 67-й порт DHCP-сервера, при передаче в обратном направлении - 68-й. Эти номера портов, как и схожая структура сообщений, обеспечивают обратную совместимость DHCP с BOOTP. Конкретные процедуры взаимодействия клиентов и серверов BOOTP и DHCP регламентирует документ RFC 1542.

Выдача адреса в аренду производится по запросу клиента. DHCP-сервер (или группа серверов) гарантирует, что выделенный адрес до истечения срока его аренды не будет выдан другому клиенту; при повторных обращениях сервер старается предложить клиенту адрес, которым тот пользовался ранее. Со своей стороны, клиент может запросить пролонгацию срока аренды адреса либо, наоборот, досрочно отказаться от него. Протоколом предусмотрена также выдача IP-адреса в

неограниченное пользование. При острой нехватке адресов сервер может сократить срок аренды адреса по сравнению с запрошенным.

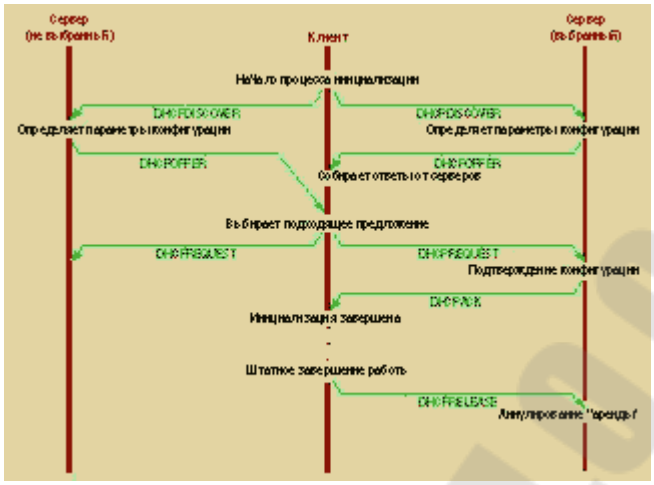


Рисунок 35 – Последовательность событий при выделении IP-адреса

Выдача нового адреса. Последовательность событий в этом случае такова (рис. 35).

1. Клиент посылает в собственную физическую подсеть широковещательное сообщение DHCPDISCOVER, в котором могут указываться устраивающие клиента IP-адрес и срок его аренды. Если в данной подсети DHCP-сервер отсутствует, сообщение будет передано в другие подсети ретранслирующими агентами протокола BOOTP (они же вернут клиенту ответные сообщения сервера).

2. Любой из DHCP-серверов может ответить на поступившее сообщение DHCPDISCOVER сообщением DHCPOFFER, включив в него доступный IP-адрес (yiaddr) и, если требуется, параметры конфигурации клиента. На этой стадии сервер не обязан резервировать указанный адрес. В принципе, он имеет право предложить его другому клиенту, также отправившему запрос DHCPDISCOVER. Тем не менее спецификации RFC 2131 рекомендуют серверу без необходимости не применять подобную тактику, а кроме того, убедиться (например, выдав эхо-запрос ICMP) в том, что предложенный адрес в текущий момент не используется каким-либо из компьютеров сети.

3. Клиент не обязан реагировать на первое же поступившее предложение. Допускается, чтобы он дождался откликов от нескольких серверов и, остановившись на одном из предложений, отправил в сеть широковещательное сообщение DHCPREQUEST. В нем содержатся идентификатор выбранного сервера и, возможно, желательные значения запрашиваемых параметров конфигурации.

Не исключено, что клиента не устроит ни одно из серверных предложений. Тогда вместо DHCPREQUEST он снова выдаст в сеть запрос DHCPDISCOVER, а серверы так и не узнают, что их предложения отклонены. Именно по этой причине сервер не обязан резервировать помещенный в DHCPOFFER адрес.

Если в процессе ожидания серверных откликов на DHCPDISCOVER достигнут тайм-аут, клиент выдает данное сообщение повторно.

4. Присутствующий в сообщении DHCPREQUEST идентификатор позволяет соответствующему DHCP-серверу убедиться в том, что клиент принял именно его предложение. В ответ сервер отправляет подтверждение DHCPACK, содержащее значения требуемых параметров конфигурации, и производит соответствующую запись в базу данных.

Если к моменту поступления сообщения DHCPREQUEST предложенный адрес уже <ушел> к другому клиенту (например, первая станция слишком долго <размышляла> над поступившими предложениями), сервер отвечает сообщением DHCPNACK.

5. Получив сообщение DHCPACK, клиент обязан убедиться в уникальности IP-адреса (средствами протокола ARP) и зафиксировать суммарный срок его аренды. Последний рассчитывается как время, прошедшее между отправкой сообщения DHCPREQUEST и приемом ответного сообщения DHCPACK, плюс срок аренды, указанный в DHCPACK.

Обнаружив, что адрес уже используется другой станцией, клиент обязан отправить серверу сообщение DHCPDECLINE и не ранее чем через 10 с начать всю процедуру снова. Процесс конфигурирования возобновляется и при получении серверного сообщения DHCPNACK.

При достижении тайм-аута в процессе ожидания серверных откликов на сообщение DHCPREQUEST клиент выдает его повторно.

6. Для досрочного прекращения аренды адреса клиент отправляет серверу сообщение DHCPRELEASE.

Приведенная последовательность действий заметно упрощается, если станция-клиент желает повторно работать с IP-адресом, который когда-то уже был ей выделен. В этом случае первым отправляемым сообщением является DHCPREQUEST, в котором клиент указывает прежде использовавшийся адрес. В ответ он может получить сообщение DHCPACK или DHCPNACK (если адрес занят либо клиентский запрос является некорректным, например из-за перемещения клиента в другую подсеть). Обязанность проверить уникальность IP-адреса опять-таки возлагается на клиента.

Выбор адреса DHCP-сервером. Если на момент получения запроса DHCPDISCOVER сервер не располагает свободными IP-адресами, он может направить уведомление о возникшей проблеме администратору. В противном случае при выборе адреса обычно применяется следующий алгоритм. Клиенту выделяется адрес, записанный за ним в данный момент. Если это невозможно, сервер предложит адрес, которым пользовался клиент до окончания срока последней аренды (при условии, что данный адрес свободен), либо адрес, запрошенный самим клиентом при помощи соответствующей опции (опять же, если адрес не занят). Наконец, в том случае, когда все предыдущие варианты не

проходят, новый адрес выбирается из пула доступных адресов с учетом подсети, из которой поступил клиентский запрос.

Заметим, что исходя из определенной сетевым администратором политики сервер может выдать клиенту адрес, отличающийся от запрошенного (даже при доступности последнего), вообще отказать в предоставлении адреса или предложить адрес, относящийся к другой подсети. Более того, DHCP-сервер вообще не обязан реагировать на каждый поступивший запрос DHCPDISCOVER. Это предоставляет администратору возможность контролировать доступ к сети, например разрешив серверу отвечать только тем клиентам, которые предварительно зарегистрировались с помощью специальной процедуры.

Истечение срока аренды. По мере того как срок аренды подходит к концу, клиент может завершить работу с данным адресом, отправив на DHCP-сервер сообщение DHCPRELEASE, либо заблаговременно запросить продление срока аренды. В первом случае возвращение в сеть потребует выполнения всей процедуры инициализации заново. Во втором - станция продолжит функционировать в сети без видимого замедления работы пользовательских приложений.

При пролонгировании аренды клиент проходит два состояния - обновления адреса (RENEWING) и обновления конфигурации (REBINDING). Первое наступает примерно на половине срока аренды адреса (так называемый момент T1), второе - по истечении приблизительно 7/8 полного времени аренды (момент T2); для рассинхронизации процессов реконфигурирования разных клиентов значения этих временных меток рандомизируются с помощью случайной добавки.

В момент T1 клиент оправляет DHCP-серверу, выдавшему адрес, сообщение DHCPREQUEST с просьбой продлить срок аренды. Получив положительный ответ (DHCPACK), клиент пересчитывает срок аренды и продолжает работу в обычном режиме. Клиент ожидает прихода ответа от сервера в течение (T2 - t)/2 с (при условии, что это значение не меньше 60 с), где t - время отсылки последнего сообщения DHCPREQUEST, после чего отправляет данное сообщение повторно.

Если ответ от сервера не поступил к моменту T2, клиент переходит в состояние REBINDING и передает уже широковещательное сообщение DHCPREQUEST со своим текущим сетевым адресом. В этом случае моменты повторных выдач запросов DHCPREQUEST рассчитываются аналогично предыдущему случаю, только вместо T2 фигурирует время окончания срока аренды.

Не исключено, однако, что ответ DHCPACK не придет до окончания срока аренды. Тогда клиент обязан немедленно прекратить выполнение любых сетевых операций и заново начать процесс инициализации. Если запоздавший ответ DHCPACK все-таки поступит, клиенту рекомендуется сразу же возобновить работу под прежним адресом.

Параметры конфигурации

Хранение параметров сетевой конфигурации станций-клиентов является второй услугой, предоставляемой DHCP-сервером. В создаваемой базе данных на

каждого клиента заводится отдельная запись с уникальным ключом-идентификатором и строкой конфигурационных параметров.

Роль идентификатора может играть пара <номер подсети IP, аппаратный адрес>, которая позволит использовать аппаратный адрес сразу в нескольких подсетях, либо пара <номер подсети IP, имя хост-компьютера>, позволяющая серверу взаимодействовать с клиентом, перемещенным в другую подсеть.

Что касается собственно параметров конфигурации, то их набор, поддерживаемый протоколом DHCP, определен в спецификациях RFC 1122, 1123, 1196 и 1256. В него входят выданный адрес, срок его аренды, назначавшиеся ранее адреса, а также максимальный размер реассемблируемого пакета, перечень фильтров для нелокальной маршрутизации от источника, адрес, используемый в широковещательных пакетах, параметры статических маршрутов и т.д. Впрочем, из всей совокупности допустимых параметров (а их более 30) в процессе инициализации могут передаваться только те, которые действительно необходимы для работы клиента либо определяются спецификой конкретной подсети.

Редукция объема передаваемых сведений о конфигурации достигается двумя способами. Во-первых, для большей части параметров в упомянутых выше документах RFC определены значения, принимаемые по умолчанию. Клиент будет использовать их, если в сообщении, поступившем от сервера, какие-то параметры опущены. Во-вторых, отправляя сообщение DHCPDISCOVER или DHCPREQUEST, клиентская станция может явно указать в нем параметры, значения которых она хотела бы получить.

Очевидно, что в обоих случаях передача параметров конфигурации осуществляется в ходе основной процедуры выделения IP-адреса. Возможен, однако, случай, когда клиент уже имеет IP-адрес (например, он был задан вручную). Тогда он может выдать сообщение DHCPINFORM*, содержащее уже имеющийся адрес и запрос об отдельных параметрах конфигурации. Получив это сообщение, DHCP-сервер проверяет правильность адреса клиента (но не наличие аренды) и направляет ему сообщение DHCPACK с требуемыми параметрами конфигурации.

Отметим одно логическое противоречие, с которым связано применение протокола DHCP. Алгоритм выделения IP-адреса компьютеру сети предполагает, что установленное на нем программное обеспечение TCP/IP в состоянии воспринимать адресованные ему посредством <аппаратного> адреса IP-пакеты и транслировать их на IP-уровень еще до того, как станция получит свой IP-адрес, а сами средства TCP/IP будут полностью сконфигурированы. Такая возможность, очевидно, существует не всегда. Для работы с клиентами, не способными корректно обрабатывать одноадресные IP-дейтаграммы, используется поле flags. Такие клиенты должны установить первый бит данного поля в единичное значение, тем самым указав серверу на необходимость отправки в соответствующую подсеть только широковещательных сообщений.

Недостатки DHCP

Освобождая сетевых администраторов от множества рутинных операций, DHCP оставляет нерешенными ряд проблем, которые рано или поздно могут возникнуть в реальной сетевой среде.

К недостаткам этого протокола прежде всего следует отнести крайне низкий уровень информационной безопасности, что обусловлено непосредственным использованием протоколов UDP и IP. В настоящее время не существует практически никакой защиты от появления в сети несанкционированных DHCP-серверов, способных рассылать клиентам ошибочную или потенциально опасную информацию - некорректные или уже задействованные IP-адреса, неверные сведения о маршрутизации и т.д. И наоборот, клиенты, запущенные с неблаговидными целями, могут извлекать конфигурационные сведения, предназначенные для <законных> компьютеров сети, и тем самым оттягивать на себя значительную часть имеющихся ресурсов. Понятно, что возможности административного ограничения доступа, о которых говорилось выше, не способны закрыть эту брешь в системе информационной безопасности.

По мнению некоторых экспертов, в настоящее время DHCP недостаточно отказоустойчив. Протоколу явно недостает механизма активного уведомления клиентов об экстремальных ситуациях (например, о систематической нехватке адресов) и серверного подтверждения об освобождении адреса, иногда в сети наблюдаются всплески числа запросов на повторное использование адресов и т.д. Впрочем, работа над протоколом еще не завершена, и не исключено, что некоторые недостатки будут устранены в последующих редакциях.

Тема 13. Основные принципы маршрутизации

Маршрутизаторы в сетевых технологиях

Объединение нескольких локальных сетей в глобальную (распределенную, составную) WAN-сеть происходит с помощью устройств и протоколов сетевого Уровня 3 семиуровневой эталонной модели или уровня межсетевого взаимодействия четырехуровневой модели TCP/IP. Если LAN объединяют рабочие станции, периферию, терминалы и другое сетевое оборудование в одной аудитории или в одном здании, то WAN обеспечивают соединение LAN на широком географическом пространстве. В составную распределенную сеть (internetwork, internet) входят как локальные сети и подсети (subnet), так и отдельные пользователи. Устройствами, объединяющими LAN в составную сеть, являются:

- маршрутизаторы (routers);
- модемы;
- коммуникационные серверы.

Наиболее распространенными устройствами межсетевого взаимодействия сетей, подсетей и устройств являются маршрутизаторы. Они представляют собой

специализированные компьютеры для выполнения специфических функций сетевых устройств. В лекции 4 было показано, что маршрутизаторы используются, чтобы сегментировать локальную сеть на широковещательные домены, т. е. являются устройствами LAN, но они применяются и как устройства формирования глобальных сетей. Поэтому маршрутизаторы имеют как LAN-, так и WAN-интерфейсы. Маршрутизаторы используют WAN-интерфейсы, чтобы связываться друг с другом, и LAN-интерфейсы – для связи с узлами (компьютерами), например через коммутаторы. Поэтому маршрутизаторы являются устройствами как локальных, так и глобальных сетей. Маршрутизаторы являются также основными устройствами больших корпоративных сетей.

На рис. 36 приведен пример того, как маршрутизаторы A, B и C объединяют несколько локальных сетей (локальные сети № 1, № 2, № 3) в распределенную (составную) сеть. Поэтому маршрутизаторы имеют интерфейсы как локальных, так и глобальных соединений. К локальным сетям, созданным на коммутаторах, маршрутизатор присоединен через интерфейсы, которые на рис. 6.1 обозначены через F0/1, что означает: интерфейс Fast Ethernet, слот 0, номер 1. Глобальные соединения на рис. 6.1 представлены последовательными или серийными (serial) интерфейсами S0/1, S0/2. Через такой же последовательный интерфейс реализовано соединение составной сети с сетью Интернет (Internet). Подобная структурная схема, включающая несколько последовательно соединенных маршрутизаторов, характерна для многих корпоративных сетей.

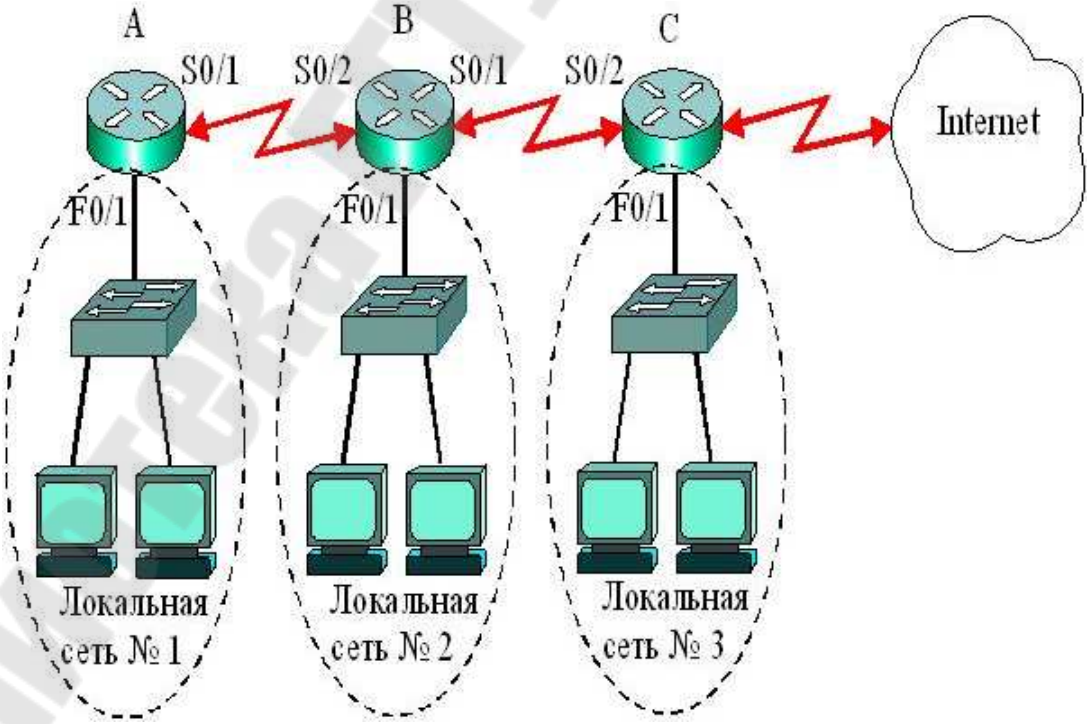


Рисунок 36 – Составная сеть на маршрутизаторах

В большинстве случаев соединение маршрутизатора локальной сети с сетью Интернет производится через сеть провайдера. Терминальное (оконечное) оборудование (Data Terminal Equipment – DTE), к которому относится и

маршрутизатор, подсоединяется к глобальной сети (или к сети провайдера) через канальное коммуникационное оборудование (Data Communications Equipment, или Data Circuit-Terminating Equipment, – DCE). Маршрутизатор обычно является оборудованием пользователя, а оборудование DCE предоставляет провайдер. Услуги, предоставляемые провайдером для терминальных устройств DTE, доступны через модем или цифровое устройство согласования с каналом связи (Channel Service Unit / Data Service Unit – CSU/DSU), которые и являются оборудованием DCE рис. 37). Оборудование DCE является ведущим в паре DCE-DTE, оно обеспечивает синхронизацию и задает скорость передачи данных.

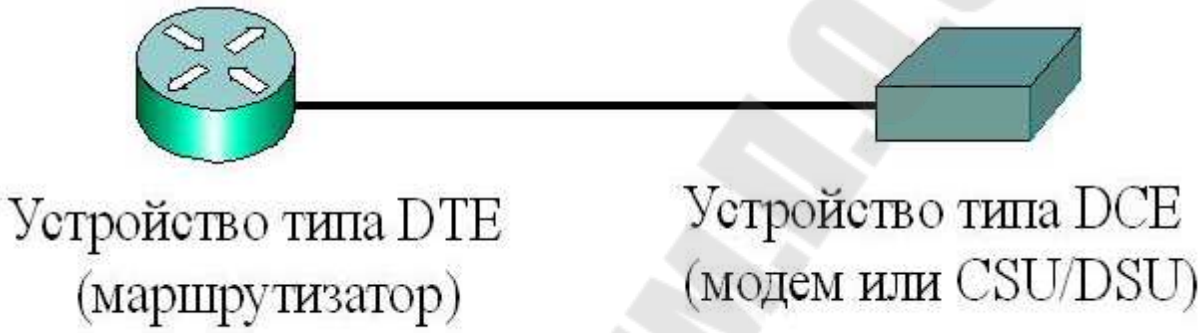


Рисунок 37 – Устройства распределенных сетей

Поскольку маршрутизаторы в распределенных сетях (рис. 36) часто соединяются последовательно, из двух последовательно соединенных серийных интерфейсов маршрутизаторов один должен выполнять роль устройства DCE, а второй – устройства DTE (рис. 38).

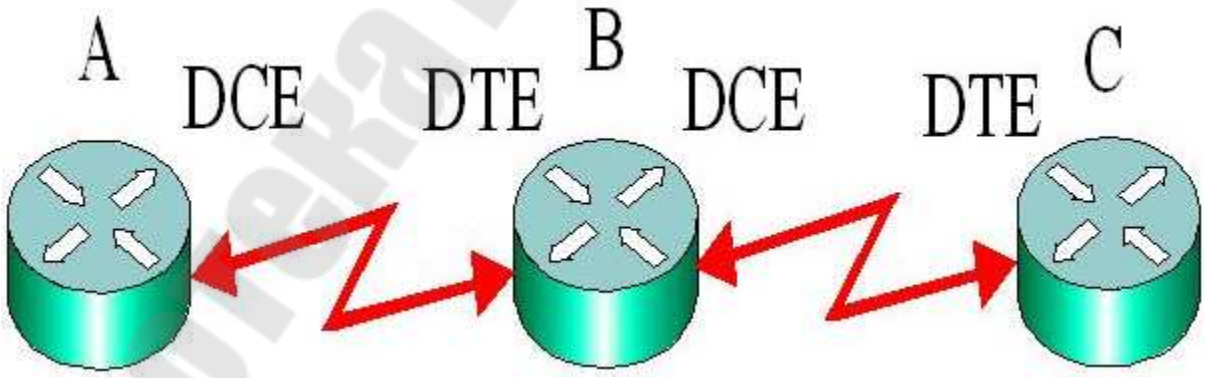


Рисунок 38 – Последовательное соединение маршрутизаторов

Главными функциями маршрутизаторов являются:

выбор наилучшего пути для пакетов к адресату назначения;

продвижение (коммутация) принятого пакета с входного интерфейса на соответствующий выходной интерфейс.

Таким образом, маршрутизаторы обеспечивают связь между сетями и определяют наилучший путь пакета данных к сети адресата, причем технологии объединяемых локальных сетей могут быть различными.

Протоколы канального (data link) уровня WAN описывают, как по сети передаются кадры. Они включают протоколы, обеспечивающие функционирование через выделенные соединения "точка-точка" и через коммутируемые соединения. Основными WAN протоколами и стандартами канального уровня являются: High-level Data Link Control (HDLC), Point-to-Point Protocol (PPP), Synchronous Data Link Control (SDLC), Serial Line Internet Protocol (SLIP), X.25, Frame Relay, ATM. Основными протоколами и стандартами физического уровня являются: EIA/TIA-232, EIA/TIA-449, V.24, V.35, X.21, G.703, EIA-530, ISDN, E1, E3, XDSL, SDH (STM-1, STM-4 и др.).

Функционируя на Уровне 3 модели OSI, маршрутизаторы принимают решения, базируясь на сетевых логических адресах (IP-адресах). Для определения наилучшего пути передачи данных через связываемые сети маршрутизаторы строят таблицы маршрутизации и обмениваются сетевой маршрутной информацией с другими маршрутизаторами. Администратор может конфигурировать статические маршруты и поддерживать таблицы маршрутизации вручную. Однако большинство таблиц маршрутизации создается и поддерживается динамически, за счет использования протоколов маршрутизации (routing protocol), которые позволяют маршрутизаторам автоматически обмениваться информацией о сетевой топологии друг с другом.

Функционирование маршрутизаторов происходит под управлением сетевой операционной системы (Internetwork Operation System – IOS), текущая (running) версия которой находится в оперативной памяти RAM (рис. 6.4). Помимо текущей версии IOS оперативная память хранит активный конфигурационный файл (Active Configuration File) и таблицы протоколов динамической маршрутизации, выполняет буферизацию пакетов и поддерживает их очередь, обеспечивает временную память для конфигурационного файла маршрутизатора, пока включено питание.

Загрузка операционной системы IOS в оперативную память обычно производится из энергонезависимой флэш-памяти (Flash), которая является перепрограммируемым запоминающим устройством (ППЗУ). После модернизации IOS она перезаписывается во флэш-память, где может храниться несколько версий. Версию операционной системы можно также сохранять на TFTP-сервере (рис. 39).

Постоянное запоминающее устройство (ПЗУ – ROM) содержит программу начальной загрузки (bootstrap) и сокращенную версию операционной системы, установленную при изготовлении маршрутизатора. Обычно эта версия IOS используется только при выходе из строя флэш-памяти. Память ROM также поддерживает команды для теста диагностики аппаратных средств (Power-On Self Test – POST).

Энергонезависимая (non-volatile) оперативная память NVRAM маршрутизатора является перепрограммируемым запоминающим устройством (ППЗУ). NVRAM хранит стартовый (startup) конфигурационный файл, который после изменения конфигурации перезаписывается в ППЗУ, где создается резервная копия (backup). Конфигурационные файлы содержат команды и параметры для управления потоком трафика, проходящим через маршрутизатор. Конфигурационный файл используется для выбора сетевых протоколов и протоколов маршрутизации, которые определяют наилучший путь для пакетов к адресуемой сети. Первоначально конфигурационный файл обычно создается с консольной линии (console) и помимо памяти NVRAM может сохраняться на TFTP-сервере (рис. 6.4). Временное хранение входящих и исходящих пакетов обеспечивается в памяти интерфейсов, которые могут быть выполнены на материнской плате или в виде отдельных модулей.

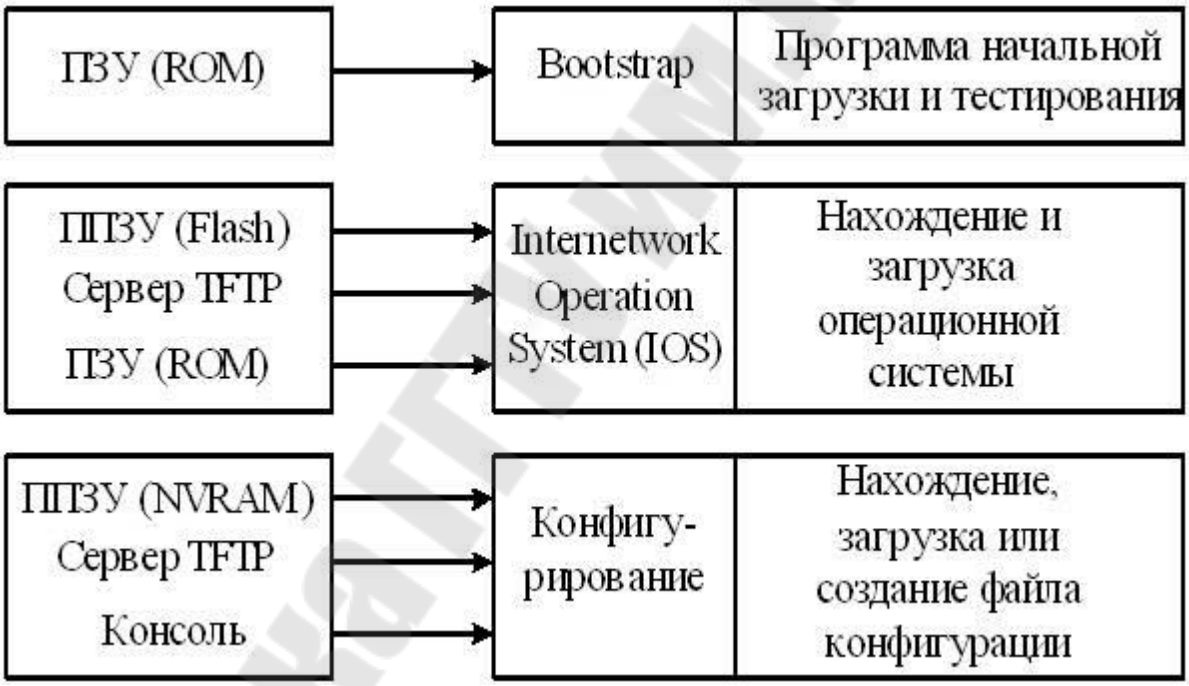


Рисунок 39 – Элементы памяти и программы маршрутизатора

При включении маршрутизатора начинает функционировать программа начальной загрузки bootstrap, которая тестирует оборудование и загружает операционную систему IOS в оперативную память RAM. В оперативную память загружается также конфигурационный файл, хранящийся в NVRAM. В процессе конфигурирования маршрутизатора задаются адреса интерфейсов, пароли, создаются таблицы маршрутизации, устанавливаются протоколы, проводится проверка параметров. Процесс коммутации и продвижения данных проходит под управлением операционной системы.

Принципы маршрутизации

Информационный поток данных, созданный на прикладном уровне, на транспортном уровне "нарезается" на сегменты, которые на сетевом уровне снабжаются заголовками и образуют пакеты. Заголовок пакета содержит сетевые IP-адреса узла назначения и узла источника. На основе этой информации средства сетевого уровня – маршрутизаторы осуществляют передачу пакетов между конечными узлами составной сети по определенному маршруту.

Маршрутизатор оценивает доступные пути к адресату назначения и выбирает наиболее рациональный маршрут на основе некоторого критерия – метрики. При оценке возможных путей маршрутизаторы используют информацию о топологии сети. Эта информация может быть сконфигурирована сетевым администратором или собрана в ходе динамического процесса обмена информацией между маршрутизаторами, который выполняется в сети протоколами маршрутизации.

Пакет, принятый на одном (входном) интерфейсе, маршрутизатор должен отправить (продвинуть) на другой (выходной) интерфейс (порт), который соответствует наилучшему пути к адресату. Чтобы передать пакеты от исходной сети (от источника) до сети адресата (назначения), на сетевом Уровне 3 маршрутизаторы используют таблицы маршрутизации для определения наиболее рационального пути.

Процесс прокладывания маршрута происходит последовательно от маршрутизатора к маршрутизатору. При прокладывании пути для пакета каждый маршрутизатор анализирует сетевую часть адреса узла назначения, заданного в заголовке поступившего пакета, т.е. вычленяет адрес сети назначения. Затем маршрутизатор обращается к таблице маршрутизации, в которой хранятся адреса всех доступных сетей, и определяет свой выходной интерфейс, на который необходимо передать (продвинуть) пакет. Таким образом, маршрутизатор ретранслирует пакет, продвигая его с входного интерфейса на выходной, для чего использует сетевую часть адреса назначения, обращаясь к таблице маршрутизации.

Выходной интерфейс связан с наиболее рациональным маршрутом к адресату. Конечный маршрутизатор на пути пакета непосредственно (прямо) связан с сетью назначения. Он использует часть сетевого адреса, содержащую адрес узла назначения, чтобы доставить пакет получателю данных.

Процесс ретрансляции пакетов маршрутизаторами рассмотрен на примере сети, приведенной на рис. 40. Маршрутизаторы в целом сетевого адреса не имеют, но поскольку они связывают между собой несколько сетей, каждый интерфейс (порт) маршрутизатора имеет уникальный адрес, сетевая часть которого совпадает с номером сети, соединенной с данным интерфейсом. Последовательные (serial) порты, соединяющие между собой маршрутизаторы, на рисунке обозначены молниевидной линией.

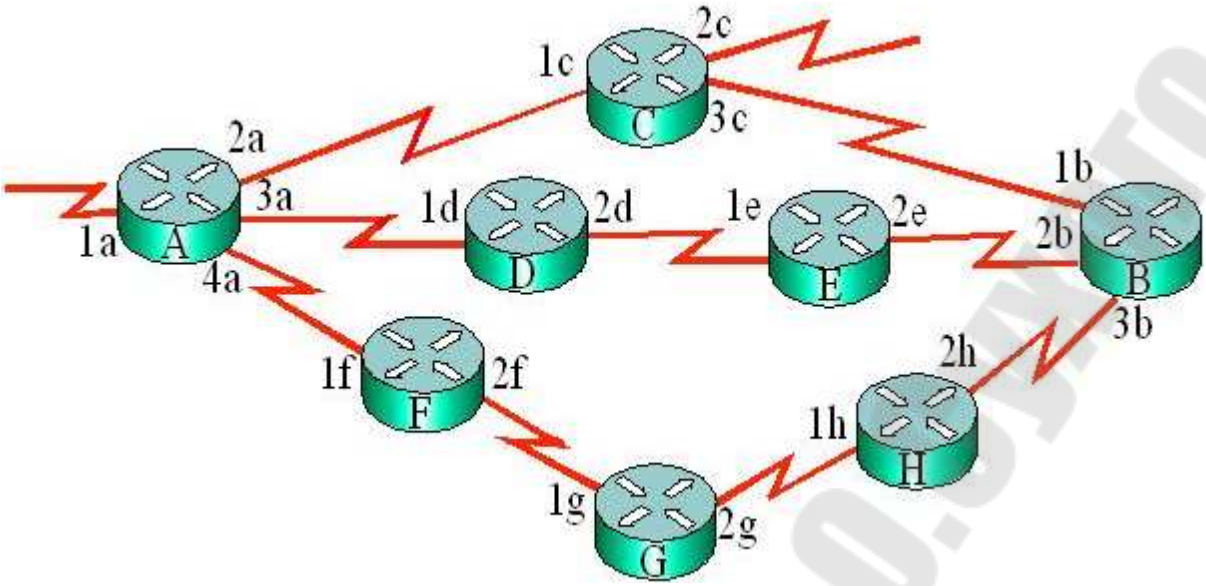


Рисунок 40 – Определения пути пакета

Путь от маршрутизатора A к маршрутизатору B может быть выбран:

- через маршрутизатор C;
- через маршрутизаторы D и E;
- через маршрутизаторы F, G и H.

Оценка наилучшего пути производится на основе метрики. Например, если метрика учитывает только количество маршрутизаторов на пути к адресату, то будет выбран первый маршрут. Если же метрика учитывает полосу пропускания линий связи, соединяющих маршрутизаторы, то может быть выбран второй или третий маршрут при условии, что на этом пути наиболее широкополосные линии связи.

При выборе первого пути функция коммутации реализуется за счет продвижения поступившего на интерфейс 1a маршрутизатора A пакета на интерфейс 2a. Таким образом, пакет попадает на интерфейс 1c маршрутизатора C, который продвинет полученный пакет на свой выходной интерфейс 3c, т. е. передаст полученный пакет маршрутизатору B.

В процессе передачи пакета по сети используются как сетевые логические адреса (IP-адреса), так и физические адреса устройств (MAC-адреса в сетях Ethernet). Например, при передаче информации с компьютера Host X локальной сети Сеть 1, (рис. 41) на компьютер Host Y, находящийся в удаленной Сети 2, определен маршрут через маршрутизаторы A, B, C.

Когда узел Host X Сети 1 передает пакет адресату Host Y из другой Сети 2, ему известен сетевой IP-адрес получателя, который записывается в заголовке пакета, т. е. известен адрес 3-го уровня. При инкапсуляции пакета в кадр источник информации Host X должен задать в заголовке кадра канальные адреса назначения и источника, т. е. адрес 2-го уровня (табл. 6).

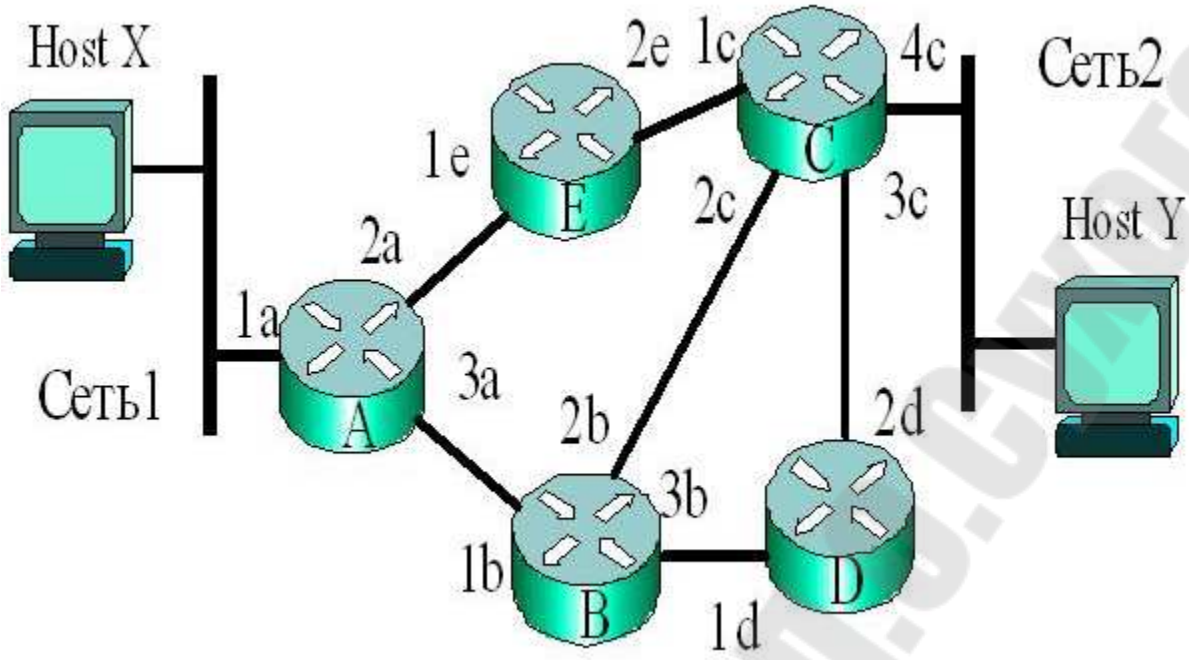


Рисунок 41 – Использование маршрутизаторов для передачи данных по сети

Таблица 6 – Основные поля кадра

Заголовок кадра		Заголовок пакета		Поле данных	Концевик (трейлер)
MAC-адрес назначения	MAC-адрес источника	IP-адрес назначения	IP-адрес источника	Данные	Контрольная сумма

У передающего узла нет информации об адресе канального уровня (MAC-адресе) узла назначения Host Y, поэтому Host X в заголовке кадра в качестве адреса назначения задаст MAC-адрес входного интерфейса 1a маршрутизатора A. Именно через этот интерфейс, называемый шлюзом по умолчанию (Default gateway), все пакеты из локальной Сети 1 будут передаваться в удаленные сети. Однако и этот адрес источнику информации Host X не известен. Процесс нахождения MAC-адреса по известному сетевому адресу реализуется с помощью протокола разрешения адресов Address Resolution Protocol – ARP, который входит в стек протоколов TCP/IP.

Протокол ARP

В локальных сетях телекоммуникаций на основе дейтаграмм устройствам необходимы как MAC-адрес, так и IP-адрес, которые для каждого узла образуют соответствующую пару. На каждом конечном узле можно посмотреть его

физический адрес и IP-адрес по команде ipconfig /all (рис. 42). Из распечатки следует, что физическим MAC-адресом конечного узла является 00-19-D1-93-7E-BE, а логическим IP-адресом – 10.0.118.52.

Протокол ARP может по IP-адресу автоматически определить MAC-адрес устройства. Каждое устройство в сети поддерживает таблицу ARP table, которая содержит соответствующие MAC- и IP-адреса других устройств той же локальной сети. Таблица ARP любого узла может быть просмотрена по команде arp -a (рис. 43). Записи таблицы хранятся в памяти RAM, где динамически поддерживаются.

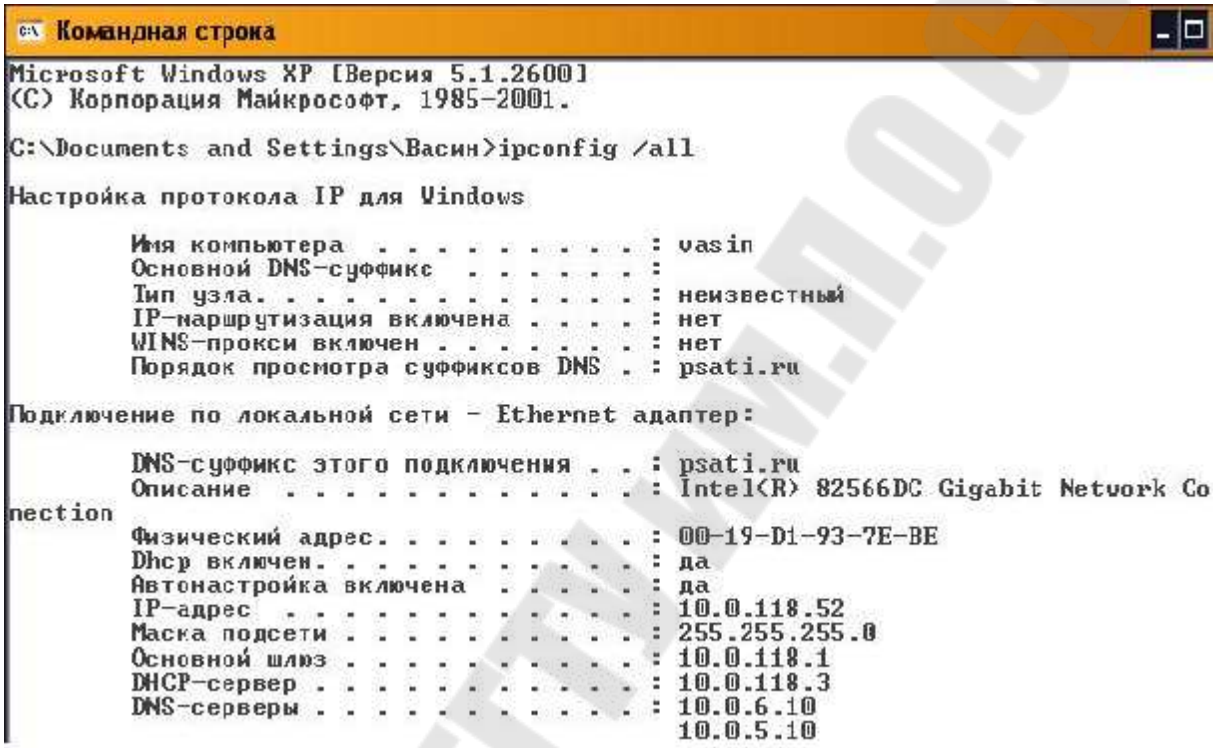


Рисунок 42 – Результат выполнения команды ipconfig /all

Если узлы долго не передают данные, то соответствующие записи из таблицы удаляются, что представлено на рис. 43, где таблица содержит только одну пару IP- и MAC-адресов.

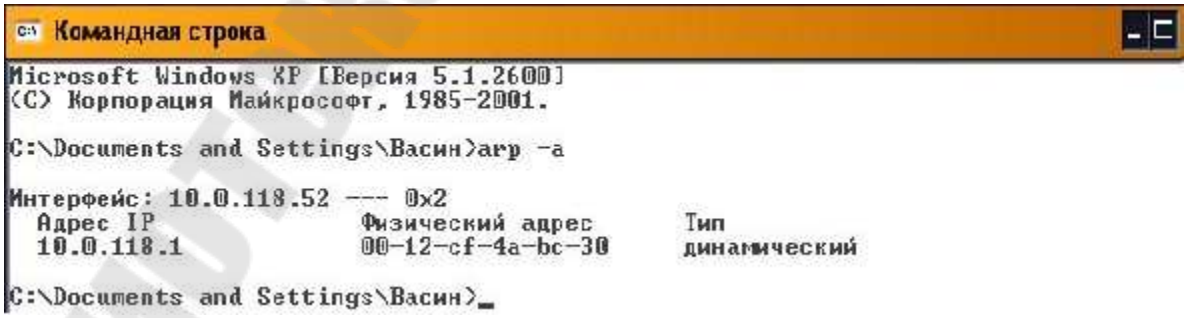


Рисунок 43 – Таблица ARP

Таблица ARP пополняется динамически путем контроля трафика локального сегмента сети. Все станции локальной сети Ethernet анализируют трафик, чтобы определить, предназначены ли данные для них. При этом IP- и MAC-адреса источников дейтаграмм записываются в таблице ARP. Например, после общения с узлом 10.0.118.65 в таблице ARP появляется вторая запись (рис. 44).

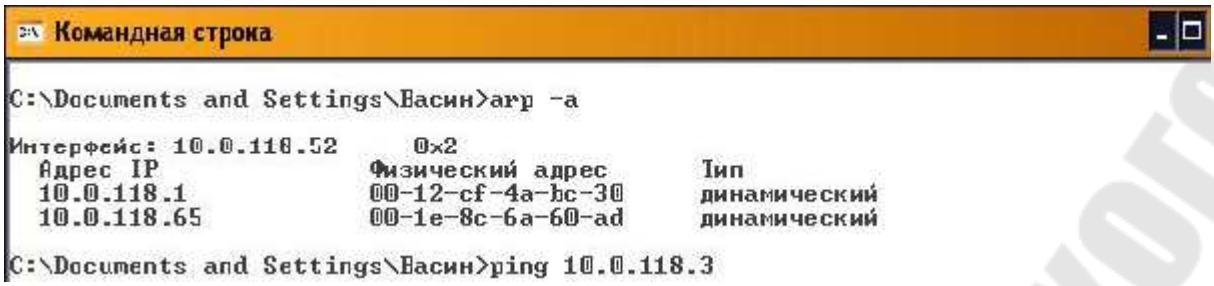


Рисунок 44 – Изменения в таблице ARP

Когда устройство передает пакет по IP-адресу назначения, оно проверяет, имеется ли в ARP-таблице соответствующий MAC-адрес назначения. Если соответствующая запись имеется, то она используется при инкапсуляции пакета в кадр данных. Данные передаются по сетевой среде, устройство назначения принимает их.

Если узел не находит соответствующей записи в таблице ARP, то он для получения MAC-адреса назначения посылает в локальную сеть широковещательный ARP-запрос, в котором задается сетевой логический IP-адрес устройства назначения. Все другие устройства сети анализируют его. Если у одного из локальных устройств IP-адрес совпадает с запрашиваемым, то устройство посылает ARP-ответ, который содержит пару IP- и MAC-адресов. Эта пара записывается в ARP-таблице. Если в локальной сети нет запрашиваемого IP-адреса, то устройство-источник сообщает об ошибке.

Когда данные передаются за пределы локальной сети, то для передачи сообщения необходимы IP- и MAC-адреса как устройства назначения, так и промежуточных маршрутизирующих устройств. Поскольку маршрутизаторы не транслируют широковещательные запросы в другие сегменты сети, в этом случае маршрутизатор в ответ на запрос посылает ARP-ответ с MAC-адресом своего входного интерфейса, на который поступил запрос. Таким образом, сформированный конечным устройством кадр поступит на интерфейс маршрутизатора, который после анализа адреса сети назначения и обращения к таблице маршрутизации продвинет пакет на выходной интерфейс.

Передать данные по адресу устройства, которое находится в другом сегменте сети, можно также за счет установки шлюза по умолчанию. Шлюз по умолчанию имеет IP-адрес входного интерфейса маршрутизатора на пути к устройству назначения. Этот адрес хранится в конфигурационном файле конечного узла (хоста). Источник сообщения сравнивает IP-адрес назначения со своим IP-адресом и определяет, находятся ли эти адреса в одном сегменте сети или в разных сегментах. Если они находятся в разных сегментах, то данные будут переданы только при условии, что установлен шлюз по умолчанию.

Таким образом, при передаче данных по сети (рис. 41) Host X для нахождения MAC-адреса назначения посылает в сеть широковещательный ARP запрос, в котором задается IP-адрес устройства назначения, на который Router A в ответ

посылает MAC-адрес своего входного интерфейса, и передаваемый пакет поступает в маршрутизатор.

Маршрутизатор A извлекает пакет из кадра, обрабатывает заголовок поступившего пакета, использует таблицу маршрутизации, чтобы определить сеть адресата, и затем продвигает пакет к выходному интерфейсу. Пакет вновь инкапсулируется в новый кадр данных и направляется следующему маршрутизатору B, при этом в заголовке кадра может указываться новый MAC-адрес входного интерфейса этого маршрутизатора. Этот процесс происходит каждый раз, когда пакет проходит через очередной маршрутизатор. В конечном маршрутизаторе (в данном примере – маршрутизатор C, рис. 41), который связан с сетью узла назначения Сеть 2, пакет инкапсулируется в кадр локальной сети адресата с MAC-адресом устройства назначения и доставляется адресату Host Y.

Для продвижения пакета к узлу назначения маршрутизатор использует таблицу маршрутизации, основными параметрами которой являются номер (адрес) сети назначения и сетевой адрес входного интерфейса следующего маршрутизатора на пути к адресату назначения. Этот адрес интерфейса получил название следующего перехода (next hop).

Таким образом, в таблице задаются:

- адрес сети назначения;
- адрес следующего перехода;
- другие дополнительные параметры, которые различаются для разных маршрутизирующих протоколов и маршрутизаторов разных фирм, производящих оборудование.

Из дополнительных параметров в таблицы маршрутизации включается информация:

- о статической или динамической маршрутизации,
- о типе используемых протоколов маршрутизации,
- о метрике, используемой при выборе возможного пути.

Принцип построения таблиц маршрутизации рассмотрен на примере сети, построенной на маршрутизаторах и коммутаторах (рис. 45). Последовательные (serial) интерфейсы маршрутизаторов на рис. 45 соединены между собой молниевидной линией, а порты Fast Ethernet – прямой линией. В приведенной схеме, например, D-f1 означает – первый Fast Ethernet порт маршрутизатора D, B-s2 – второй последовательный порт маршрутизатора B.

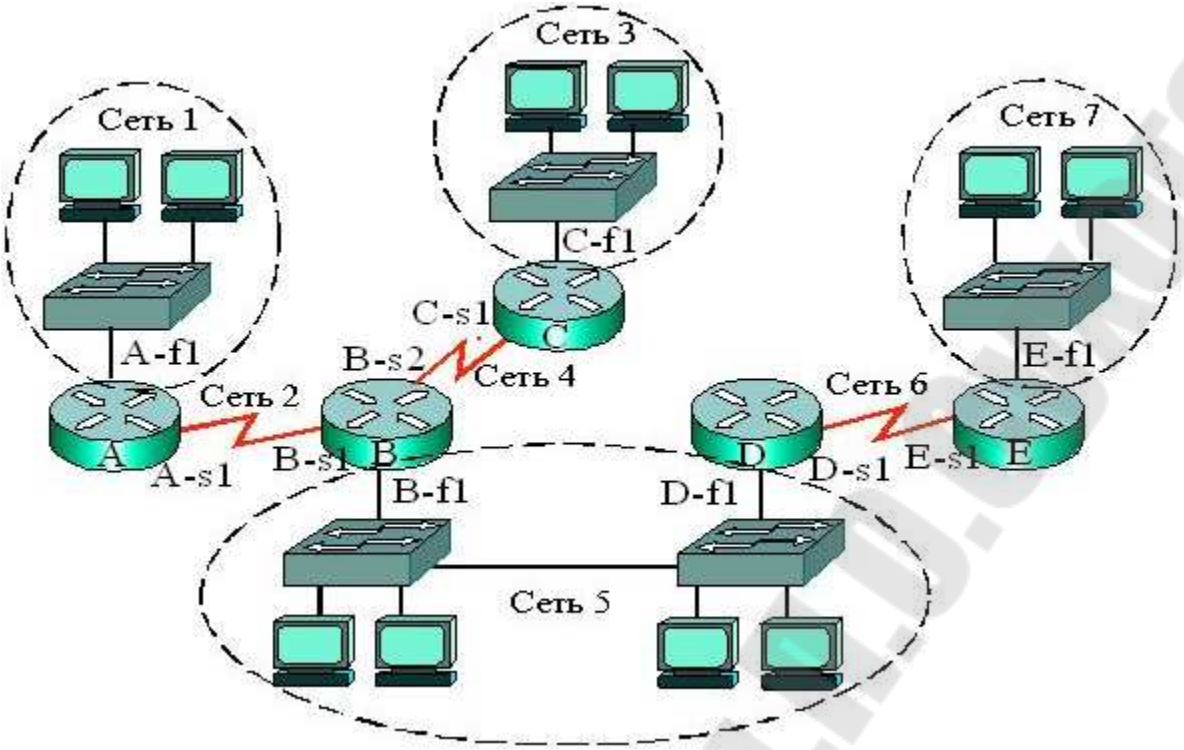


Рисунок 45 – Принцип маршрутизации в сети

Таблица маршрутизации, например маршрутизатора B (таблица 7), будет содержать информацию о маршрутах ко всем сетям (рис. 45). Маршрут к Сети 1 лежит через последовательный интерфейс A-s1 маршрутизатора A, к Сети 3 – через последовательный интерфейс C-s1 маршрутизатора C, а к сетям Сеть 6, Сеть 7 – через интерфейс D-f1 маршрутизатора D. Адреса входных интерфейсов маршрутизаторов на пути следования пакета к адресату назначения называются адресами следующего перехода (next hop).

Таблица 7 Основные параметры таблицы маршрутизации

Адрес сети назначения	Адрес следующего перехода
Сеть 1	A-s1
Сеть 3	C-s1
Сеть 6	D-f1
Сеть 7	D-f1

Вместо адреса следующего перехода часто указывают обозначение выходного интерфейса маршрутизатора, отправляющего пакет. Поскольку выходной интерфейс маршрутизатора, отправляющего пакет, и входной интерфейс следующего маршрутизатора на пути к адресату назначения соединены между собой, противоречий при этом никаких нет. Кроме удаленных сетей назначения в таблице маршрутизации указываются непосредственно (прямо) присоединенные сети с указанием выходного интерфейса. Например, таблица маршрутизации B (таблица 8) будет содержать три прямо присоединенных сети.

Таблица 8 Прямо присоединенные сети таблицы маршрутизации

Адрес присоединенной сети	Выходной интерфейс
Сеть 2	s1
Сеть 4	S2
Сеть 5	f1

Таким образом, пакет, предназначенный одному из узлов сети, например Сети 7, попав в маршрутизатор B, будет направлен на входной интерфейс D-f1 маршрутизатора D (следующий переход). В свою очередь, в таблице маршрутизации D будет задан адрес входного интерфейса E-s1 следующего маршрутизатора E, для которого Сеть 7 является непосредственно присоединенной. Поэтому маршрутизатор E направит пакет узлу назначения.

В Интернете нашли применение два основных протокола транспортного уровня, один из которых ориентирован на соединение, другой — нет. Протоколом без установления соединения является UDP. Протокол TCP, напротив, ориентирован на соединение. Так как UDP — это, на самом деле, просто IP с добавлением небольшого заголовка, мы изучим сперва его. Рассмотрим также два практических применения UDP.

Основы UDP

UDP (User Datagram Protocol — пользовательский дейтаграмм-ный протокол). UDP позволяет приложениям отправлять инкапсулированные IP-дейтаграммы без установления соединений. UDP описан в RFC 768.

С помощью протокола UDP передаются сегменты, состоящие из 8-байтного заголовка, за которым следует поле полезной нагрузки. Два номера портов служат для идентификации конечных точек внутри отправляющей и принимающей машин. Когда прибывает пакет UDP, содержимое его поля полезной нагрузки передается процессу, связанному с портом назначения. В сущности, весь смысл использования UDP вместо обычного IP заключается как раз в указании портов источника и приемника. Без этих двух полей на транспортном уровне невозможно было бы определить действие, которое следует произвести с пакетом. В соответствии с полями портов производится корректная доставка сегментов.

Информация о порте источника требуется прежде всего при создании ответа, пересылаемого отправителю. Копируя значения поля Порт источника из входящего сегмента в поле Порт назначения исходящего сегмента, процесс, посылающий ответ, может указать, какому именно процессу на противоположной стороне он предназначается.

Поле Длина UDP содержит информацию о длине сегмента, включая заголовок и полезную нагрузку. Контрольная сумма UDP не является обязательной. Если она не подсчитывается, ее значение равно 0 (настоящая нулевая контрольная сумма кодируется всеми единицами).

Отключать функцию подсчета контрольной суммы глупо, за исключением одного случая — когда нужна высокая производительность (например, при передаче оцифрованной речи).

Наверное, стоит прямо сказать о том, чего UDP не делает. Итак, UDP не занимается контролем потока, контролем ошибок, повторной передачей после приема испорченного сегмента. Все это перекладывается на пользовательские процессы. Что же он делает? UDP предоставляет интерфейс для IP путем демультиплексирования нескольких процессов, использующих порты. Это все, что он делает. Для процессов, которым хочется управлять потоком, контролировать ошибки и временные интервалы, протокол UDP — это как раз то, что доктор прописал.

Одной из областей, где UDP применяется особенно широко, является область клиент-серверных приложений. Зачастую клиент посылает короткий запрос серверу и надеется получить короткий ответ. Если запрос или ответ теряется, клиент по прошествии определенного временного интервала может попытаться еще раз. Это позволяет не только упростить код, но и уменьшить требуемое количество собщений по сравнению с протоколами, которым требуется начальная настройка.

DNS (Domain Name System — служба имен доменов) — это приложение, которое использует UDP именно так, как описано выше. В двух словах, если программе нужно найти IP-адрес по имени хоста, например, www.vanderboot.ru, она может послать UDP-пакет с этим именем на сервер DNS. Сервер в ответ на запрос посылает UDP-пакет с IP-адресом хоста. Никакой предварительной настройки не требуется, как не требуется и разрыва соединения после завершения задачи. По сети просто передаются два сообщения.

Транспортные протоколы - TCP

UDP является простым протоколом и имеет определенную область применения. В первую очередь, это клиент-серверные взаимодействия и мультимедиа. Тем не менее, большинству интернет-приложений требуется надежная, последовательная передача. UDP не удовлетворяет этим требованиям, поэтому требуется иной протокол. Такой протокол называется TCP, и он является рабочей лошадкой Интернета.

Основы TCP

Протокол TCP (Transmission Control Protocol — протокол управления передачей) был специально разработан для обеспечения надежного сквозного байтового потока по ненадежной интерсети. Объединенная сеть отличается от отдельной сети тем, что ее различные участки могут обладать сильно различающейся топологией, пропускной способностью, значениями времени задержки, размерами пакетов и другими параметрами. При разработке TCP основное внимание уделялось способности протокола адаптироваться к свойствам объединенной сети и отказоустойчивости при возникновении различных проблем.

Протокол TCP описан в RFC 793. Со временем были обнаружены различные ошибки и неточности, и по некоторым пунктам требования были изменены. Подробное описание этих уточнений и исправлений дается в RFC 1122. Расширения протокола приведены в RFC 1323.

Каждая машина, поддерживающая протокол TCP, обладает транспортной сущностью TCP, являющейся либо библиотечной процедурой, либо пользовательским процессом, либо частью ядра системы. В любом случае, транспортная сущность управляет TCP-потоками и интерфейсом с IP-уровнем. TCP-сущность принимает от локальных процессов пользовательские потоки данных, разбивает их на куски, не превосходящие 64 Кбайт (на практике это число обычно равно 460 байтам данных, что позволяет поместить их в один кадр Ethernet с заголовками IP и TCP), и посылает их в виде отдельных IP-дейтаграмм. Когда IP-дейтаграммы с TCP-данными прибывают на машину, они передаются TCP-сущности, которая восстанавливает исходный байтовый поток. Для простоты мы иногда будем употреблять «TCP» для обозначения транспортной сущности TCP (части программного обеспечения) или протокола TCP (набора правил). Из контекста будет понятно, что имеется в виду. Например, в выражении «Пользователь передает данные TCP» подразумевается, естественно, транспортная сущность TCP.

Уровень IP не гарантирует правильной доставки дейтаграмм, поэтому именно TCP приходится следить за истекшими интервалами ожидания и в случае необходимости заниматься повторной передачей пакетов. Бывает, что дейтаграммы прибывают в неправильном порядке. Восстанавливать сообщения из таких дейтаграмм обязан также TCP. Таким образом, протокол TCP призван обеспечить надежность, о которой мечтают многие пользователи и которая не предоставляется протоколом IP.

Тема 14. Передача данных по сети через сокеты

В библиотеке классов Java есть очень удобное средство, с помощью которых можно организовать взаимодействие между приложениями Java и аплетами, работающими как на одном и том же, так и на разных узлах сети TCP/IP. Это средство, родившееся в мире операционной системы UNIX, - так называемые сокеты (sockets).

Что такое сокеты?

Вы можете представить себе сокеты в виде двух розеток, в которые включен кабель, предназначенный для передачи данных через сеть. Переходя к компьютерной терминологии, скажем, что сокеты - это программный интерфейс, предназначенный для передачи данных между приложениями.

Прежде чем приложение сможет выполнять передачу аили прием данных, оно должно создать сокет, указав при этом адрес узла IP, номер порта, через который будут передаваться данные, и тип сокета.

С адресом узла IP мы уже сталкивались. Номер порта служит для идентификации приложения. Заметим, что существуют так называемые "хорошо известные" (well known) номера портов, зарезервированные для различных приложений. Например, порт с номером 80 зарезервирован для использования серверами Web при обмене данными через протокол HTTP.

Что же касается типов сокетов, то их два - потоковые и датаграммные.

С помощью потоковых сокетов вы можете создавать каналы передачи данных между двумя приложениями Java в виде потоков, которые мы уже рассматривали во второй главе. Потоки могут быть входными или выходными, обычными или форматированными, с использованием или без использования буферизации. Скоро вы убедитесь, что организовать обмен данными между приложениями Java с использованием потоковых сокетов не труднее, чем работать через потоки с обычными файлами.

Заметим, что потоковые сокеты позволяют передавать данные только между двумя приложениями, так как они предполагают создание канала между этими приложениями. Однако иногда нужно обеспечить взаимодействие нескольких клиентских приложений с одним серверным или нескольких клиентских приложений с несколькими серверными приложениями. В этом случае вы можете либо создавать в серверном приложении отдельные задачи и отдельные каналы для каждого клиентского приложения, либо воспользоваться датаграммными сокетами. Последние позволяют передавать данные сразу всем узлам сети, хотя такая возможность редко используется и часто блокируется администраторами сети.

Для передачи данных через датаграммные сокеты вам не нужно создавать канал - данные посылаются непосредственно тому приложению, для которого они предназначены с использованием адреса этого приложения в виде сокета и номера порта. При этом одно клиентское приложение может обмениваться данными с несколькими серверными приложениями или наоборот, одно серверное приложение - с несколькими клиентскими.

К сожалению, датаграммные сокеты не гарантируют доставку передаваемых пакетов данных. Даже если пакеты данных, передаваемые через такие сокеты, дошли до адресата, не гарантируется, что они будут получены в той же самой последовательности, в которой были переданы. Потоковые сокеты, напротив, гарантируют доставку пакетов данных, причем в правильной последовательности.

Причина отстутствия гарантии доставки данных при использовании датаграммных сокетов заключается в использовании такими сокетами протокола UDP, который, в свою очередь, основан на протоколе с негарантированной доставкой IP. Потоковые сокеты работают через протокол гарантированной доставки TCP.

Работа с потоковыми сокетами

Как мы уже говорили, интерфейс сокетов позволяет передавать данные между двумя приложениями, работающими на одном или разных узлах сети. В процессе создания канала передачи данных одно из этих приложений выполняет роль сервера, а другое - роль клиента. После того как канал будет создан, приложения становятся равноправными - они могут передавать друг другу данные симметричным образом.

Рассмотрим этот процесс в деталях.

Инициализация сервера

Вначале мы рассмотрим действия приложения, которое на момент инициализации является сервером.

Первое, что должно сделать серверное приложение, это создать объект класса ServerSocket, указав конструктору этого класса номер используемого порта:

```
ServerSocket ss;
ss = new ServerSocket(9999);
```

Заметим, что объект класса ServerSocket вовсе не является сокетом. Он предназначен всего лишь для установки канала связи с клиентским приложением, после чего создается сокет класса Socket, пригодный для передачи данных.

Установка канала связи с клиентским приложением выполняется при помощи метода accept, определенного в классе ServerSocket:

```
Socket s;
s = ss.accept();
```

Метод accept приостанавливает работу вызвавшего потока до тех пор, пока клиентское приложение не установит канал связи с сервером. Если ваше приложение однопоточное, его работа будет блокирована до момента установки канала связи. Избежать полной блокировки приложения можно, если выполнять создание канала передачи данных в отдельном потоке.

Как только канал будет создан, вы можете использовать сокет сервера для образования входного и выходного потока класса InputStream и OutputStream, соответственно:

```
InputStream is;
OutputStream os;
is = s.getInputStream();
os = s.getOutputStream();
```

Эти потоки можно использовать таким же образом, что и потоки, связанные с файлами.

Обратите также внимание на то, что при создании серверного сокета мы не указали адрес IP и тип сокета, ограничившись только номером порта.

Что касается адреса IP, то он, очевидно, равен адресу IP узла, на котором запущено приложение сервера. В классе ServerSocket определен метод getInetAddress, позволяющий определить этот адрес:

public InetAddress getInetAddress();

Тип сокета указывать не нужно, так как для работы с датаграммными сокетами предназначен класс DatagramSocket, который мы рассмотрим позже.

Инициализация клиента

Процесс инициализации клиентского приложения выглядит весьма просто. Клиент должен просто создать сокет как объект класса Socket, указав адрес IP серверного приложения и номер порта, используемого сервером:

```
Socket s;
s = new Socket("localhost",9999);
```

Здесь в качестве адреса IP мы указали специальный адрес localhost, предназначенный для тестирования сетевых приложений, а в качестве номера порта - ззначение 9999, использованное сервером.

Теперь можно создавать входной и выходной потоки. На стороне клиента эта операция выполняется точно также, как и на стороне сервера:

```
InputStream is;
OutputStream os;
is = s.getInputStream();
os = s.getOutputStream();
```

Передача данных между клиентом и сервером

После того как серверное и клиентское приложения создали потоки для приема и передачи данных, оба этих приложения могут читать и писать в канал данных, вызывая методы read и write, определенные в классах InputStream и OutputStream.

Ниже представлен фрагмент кода, в котором приложение вначале читает данные из входного потока в буфер buf, а затем записывает прочитанные данные в выходной поток:

```
byte buf[] = new byte[512];
int lenght;
lenght = is.read(buf);
os.write(buf, 0, lenght);
os.flush();
```

На базе потоков класса InputStream и OutputStream вы можете создать буферизованные потоки и потоки для передачи форматированных данных, о которых мы рассказывали раньше.

Завершение работы сервера и клиента

После завершения передачи данных вы должны закрыть потоки, вызвав метод close:

Яis.close();
os.close();

Когда канал передачи данных больше не нужен, сервер и клиент должны закрыть сокет, вызвав метод close, определенный в классе Socket:

s.close();

Серверное приложение, кроме того, должно закрыть соединение, вызвав метод close для объекта класса ServerSocket:

s*s.close()*;

Конструкторы класса Socket

Чаще всего для создания сокетов в клиентских приложениях вы будете использовать один из двух конструкторов, прототипы которых приведены ниже:

public Socket(String host,int port);
public Socket(InetAddress address,int port);

Первый из этих конструкторов позволяет указывать адрес серверного узла в виде текстовой строки, второй - в виде ссылки на объект класса InetAddress. Вторым параметром задается номер порта, с использованием которого будут передаваться данные.

В классе Socket определена еще одна пара конструкторов, которая, однако не рекомендуется для использования:

public Socket(String host,
int port, boolean stream);
public Socket(InetAddress address,
int port, boolean stream);

В этих конструкторах последний параметр определяет тип сокета. Если этот параметр равен true, создается потоковый сокет, а если false - датаграммный. Заметим, что для работы с датаграммными сокетами следует использовать класс DatagramSocket.

Методы класса Socket

Перечислим наиболее интересные, на наш взгляд, методы класса Socket.

Прежде всего, это методы getInputStream и getOutputStream, предназначенные для создания входного и выходного потока, соответственно:

public InputStream getInputStream();
public OutputStream getOutputStream();

Эти потоки связаны с сокетом и должны быть использованы для передачи данных по каналу связи.

Методы getInetAddress и getPort позволяют определить адрес IP и номер порта, связанные с данным сокетом (для удаленного узла):

public InetAddress getInetAddress();
public int getPort();

Метод getLocalPort возвращает для данного сокета номер локального порта:

public int getLocalPort();

После того как работа с сокетом завершена, его необходимо закрыть методом close:

public void close();

И, наконец, метод toString возвращает текстовую строку, представляющую сокет:

public String toString();

Использование датаграммных сокетов

Как мы уже говорили, датаграммные сокеты не гарантируют доставку пакетов данных. Тем не менее, они работают быстрее потоковых и обеспечивают возможность широковещательной расслыки пакетов данных одновременно всем узлам сети. Последняя возможность используется не очень широко в сети Internet, однако в корпоративной сети Intranet вы вполне можете ей воспользоваться.

Для работы с датаграммными сокетами приложение должно создать сокет на базе класса DatagramSocket, а также подготовить объект класса DatagramPacket, в который будет записан принятый от партнера по сети блок данных.

Канал, а также входные и выходные потоки создавать не нужно. Данные передаются и принимаются методами send и receive, определенными в классе DatagramSocket.

Класс DatagramSocket

Рассмотрим конструкторы и методы класса DatagramSocket, предназначенного для сояяяяздания и использования датаграммных сокетов.

В классе DatagramSocket определены два конструктора, прототипы которых представлены ниже:

public DatagramSocket(int port);
public DatagramSocket();

Первый из этих конструкторов позволяет определить порт для сокета, второй предполагает использование любого свободного порта.

Обычно серверные приложения работают с использованием какого-то заранее определенного порта, номер которого известен клиентским приложениям. Поэтому для серверных приложений больше подходит первый из приведенных выше конструкторов.

Клиентские приложения, напротив, часто применяют любые свободные на локальном узле порты, поэтому для них годится конструктор без параметров.

Кстати, с помощью метода getLocalPort приложение всегда может узнать номер порта, закрепленного за данным сокетом:

public int getLocalPort();

Прием и передача данных на датаграммном сокете выполняется с помощью методов receive и send, соответственно:

public void receive(DatagramPacket p);
public void send(DatagramPacket p);

В качестве параметра этим методам передается ссылка на пакет данных (соответственно, принимаемый и передаваемый), определенный как объект класса DatagramPacket. Этот класс будет рассмотрен позже.

Еще один метод в классе DatagramSocket, которым вы будете пользоваться, это метод close, предназначенный для закрытия сокета:

public void close();

Напомним, что сборка мусора в Java выполняется только для объектов, находящихся в оперативной памяти. Такие объекты, как потоки и сокеты, вы должны закрывать после использования самостоятельно.

Класс DatagramPacket

Перед тем как принимать или передавать данные с использованием методов receive и send вы должны подготовить объекты класса DatagramPacket. Метод receive запишет в такой объект принятые данные, а метод send - перешлет данные из объекта класса DatagramPacket узлу, адрес которого указан в пакете.

Подготовка объекта класса DatagramPacket для приема пакетов выполняется с помощью следующего конструктора:

public DatagramPacket(byte ibuf[],
int ilength);

Этому конструктору передается ссылка на массив ibuf, в который нужно будет записать данные, и размер этого массива ilength.

Если вам нужно подготовить пакет для передачи, воспользуйтесь конструктором, который дополнительно позволяет задать адрес IP iaddr и номер порта iport узла назначения:

public DatagramPacket(byte ibuf[],
int ilength,
InetAddress iaddr, int iport);

Таким образом, информация о том, в какой узел и на какой порт необходимо доставить пакет данных, хранится не в сокете, а в пакете, то есть в объекте класса DatagramPacket.

Помимо только что описанных конструкторов, в классе DatagramPacket определены четыре метода, позволяющие получить данные и информацию об адресе узла, из которого пришел пакет, или для которого предназначен пакет.

Метод getData возвращает ссылку на массив данных пакета:

public byte[] getData();

Размер пакета, данные из которого хранятся в этом массиве, легко определить с помощью метода getLength:

public int getLength();

Методы getAddress и getPort позволяют определить адрес и номер порта узла, откуда пришел пакет, или узла, для которого предназначен пакет:

public InetAddress getAddress();
public int getPort();

Если вы создаете клиент-серверную систему, в которой сервер имеет заранее известный адрес и номер порта, а клиенты - произвольные адреса и различные номера портов, то после получения пакета от клиента сервер может определить с помощью методов getAddress и getPort адрес клиента для установления с ним связи.

Если же адрес сервера неизвестен, клиент может посылать широковещательные пакеты, указав в объекте класса DatagramPacket адрес сети. Такая методика обычно используется в локальных сетях.

Адрес IP состоит из двух частей - адреса сети и адреса узла. Для разделения компонент 32-разрядного адреса IP используется 32-разрядная маска, в которой битам адреса сети соответствуют единицы, а битам адреса узла - нули.

Например, адрес узла может быть указан как 193.24.111.2. Исходя из значения старшего байта адреса, это сеть класса С, для которой по умолчанию используется маска 255.255.255.0. Следовательно, адрес сети будет такой: 193.24.111.0.

РАЗДЕЛ 5. СРЕДСТВА ОПЕРАЦИОННЫХ СИСТЕМ ДЛЯ РАБОТЫ С КОМПЬЮТЕРНЫМИ СЕТЯМИ

Тема 15. Сетевые операционные системы

Операционную систему компьютера часто определяют как взаимосвязанный набор системных программ, который обеспечивает эффективное управление ресурсами компьютера (памятью, процессором, внешними устройствами, файлами и др.), а также предоставляет пользователю удобный интерфейс для работы с аппаратурой компьютера и разработки приложений. Говоря о сетевых ОС, мы, очевидно, должны расширить границы управляемых ресурсов за пределы одного компьютера.

Сетевой операционной системой (ОС) называют операционную систему компьютера, которая помимо управления локальными ресурсами предоставляет пользователям и приложениям возможность эффективного и удобного доступа к информационным и аппаратным ресурсам других компьютеров сети.

Сегодня практически все операционные системы являются сетевыми.

В сетевых ОС удаленный доступ к сетевым ресурсам обеспечивается:

- сетевыми службами;
- средствами транспортировки сообщений по сети (в простейшем случае — сетевыми интерфейсными картами и их драйверами).

Функции сетевых ОС:

- управление каталогами и файлами;
- управление ресурсами;
- коммуникационные функции;
- защита от несанкционированного доступа;
- обеспечение отказоустойчивости;
- управление сетью.

Управление каталогами и файлами является одной из первоочередных функций сетевой операционной системы, обслуживаемых специальной сетевой файловой подсистемой. Пользователь получает от этой подсистемы возможность обращаться к файлам, физически расположенным в сервере или в другой станции данных, применяя привычные для локальной работы языковые средства. При обмене файлами должен быть обеспечен необходимый уровень конфиденциальности обмена (секретности данных).

Управление ресурсами включает запросы и предоставление ресурсов.

Коммуникационные функции обеспечивают адресацию, буферизацию, маршрутизацию.

Защита от несанкционированного доступа возможна на любом из следующих уровней: ограничение доступа в определенное время, и (или) для определенных станций, и (или) определенное число раз; ограничение совокупности доступных конкретному пользователю директорий; ограничение для конкретного пользователя списка возможных действий (например, только чтение файлов); пометка файлов символами типа «только чтение», «скрытность при просмотре списка файлов».

Отказоустойчивость определяется наличием в сети автономного источника питания, отображением или дублированием информации в дисковых накопителях. Отображение заключается в хранении двух копий данных на двух дисках, подключенных к одному контроллеру, а дублирование означает подключение каждого из этих двух дисков к разным контроллерам. Сетевая ОС, реализующая дублирование дисков, обеспечивает более высокий уровень отказоустойчивости.

Дальнейшее повышение отказоустойчивости связано с дублированием серверов.

Структура сетевой ОС

Сетевая операционная система составляет основу любой вычислительной сети. Каждый компьютер в сети в значительной степени автономен, поэтому под сетевой операционной системой в широком смысле понимается совокупность операционных систем отдельных компьютеров, взаимодействующих с целью обмена сообщениями и разделения ресурсов по единым правилам - протоколам. В узком смысле сетевая ОС - это операционная система отдельного компьютера, обеспечивающая ему возможность работать в сети.

Рисунок 46 – Структура сетевой ОС

В сетевой операционной системе отдельной машины можно выделить несколько частей (рис. 46):

- средства управления локальными ресурсами компьютера: функции распределения оперативной памяти между процессами, планирования и диспетчеризации процессов, управления процессорами в мультипроцессорных машинах, управления периферийными устройствами и другие функции управления ресурсами локальных ОС.

- средства предоставления собственных ресурсов и услуг в общее пользование - серверная часть ОС (сервер). Эти средства обеспечивают, например, блокировку файлов и записей, что необходимо для их совместного использования; ведение справочников имен сетевых ресурсов; обработку запросов удаленного доступа к собственной файловой системе и базе данных; управление очередями запросов удаленных пользователей к своим периферийным устройствам.

- средства запроса доступа к удаленным ресурсам и услугам и их использования - клиентская часть ОС (редиректор). Эта часть выполняет распознавание и перенаправление в сеть запросов к удаленным ресурсам от приложений и пользователей, при этом запрос поступает от приложения в локальной форме, а передается в сеть в другой форме, соответствующей требованиям сервера. Клиентская часть также осуществляет прием ответов от серверов и преобразование их в локальный формат, так что для приложения выполнение локальных и удаленных запросов неразличимо.

- коммуникационные средства ОС, с помощью которых происходит обмен сообщениями в сети. Эта часть обеспечивает адресацию и буферизацию сообщений, выбор маршрута передачи сообщения по сети, надежность передачи и т.п., то есть является средством транспортировки сообщений.

В зависимости от функций, возлагаемых на конкретный компьютер, в его операционной системе может отсутствовать либо клиентская, либо серверная части.

На рисунке 47 показано взаимодействие сетевых компонентов. Здесь компьютер 1 выполняет роль "чистого" клиента, а компьютер 2 - роль "чистого" сервера, соответственно на первой машине отсутствует серверная часть, а на второй - клиентская. На рисунке отдельно показан компонент клиентской части - редиректор. Именно редиректор перехватывает все запросы, поступающие от приложений, и анализирует их. Если выдан запрос к ресурсу данного компьютера, то он переадресовывается соответствующей подсистеме локальной ОС, если же это запрос к удаленному ресурсу, то он переправляется в сеть. При этом клиентская часть преобразует запрос из локальной формы в сетевой формат и передает его транспортной подсистеме, которая отвечает за доставку сообщений указанному серверу. Серверная часть операционной системы компьютера 2 принимает запрос, преобразует его и передает для выполнения своей локальной ОС. После того, как результат получен, сервер обращается к транспортной подсистеме и направляет ответ клиенту, выдавшему запрос. Клиентская часть преобразует результат в соответствующий формат и адресует его тому приложению, которое выдало запрос.

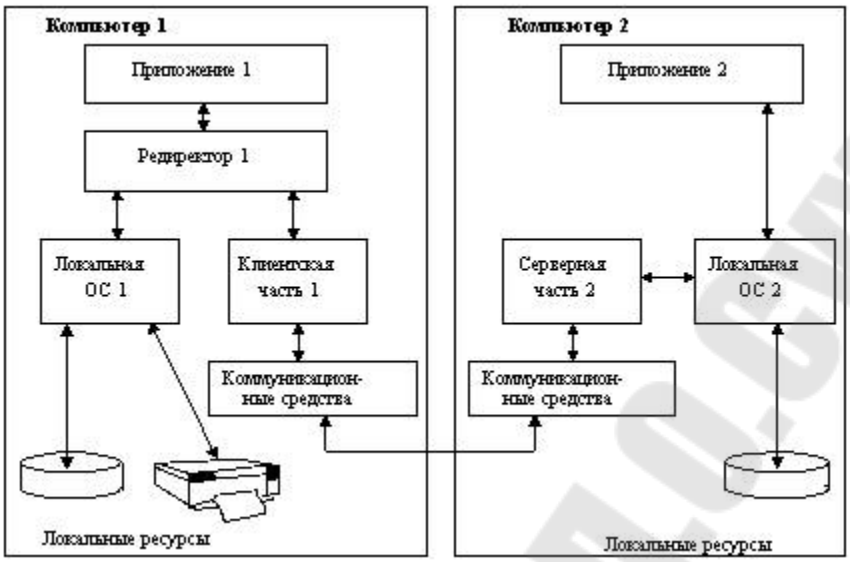


Рисунок 47 – Взаимодействие компонентов операционной системы при взаимодействии компьютеров

На практике сложилось несколько подходов к построению сетевых операционных систем (рис. 48).

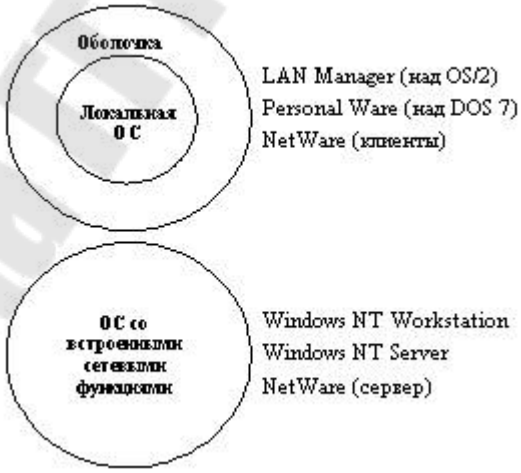


Рисунок 48 – Варианты построения сетевых ОС

Первые сетевые ОС представляли собой совокупность существующей локальной ОС и надстроенной над ней сетевой оболочки. При этом в локальную ОС встраивался минимум сетевых функций, необходимых для работы сетевой оболочки, которая выполняла основные сетевые функции. Примером такого подхода является использование на каждой машине сети операционной системы MS DOS (у которой начиная с ее третьей версии появились такие встроенные функции, как блокировка файлов и записей, необходимые для совместного доступа к файлам).

Принцип построения сетевых ОС в виде сетевой оболочки над локальной ОС используется и в современных ОС, таких, например, как LANtastic или Personal Ware.

Однако более эффективным представляется путь разработки операционных систем, изначально предназначенных для работы в сети. Сетевые функции у ОС такого типа глубоко встроены в основные модули системы, что обеспечивает их логическую стройность, простоту эксплуатации и модификации, а также высокую производительность. Примером такой ОС является система Windows NT фирмы Microsoft, которая за счет встроенности сетевых средств обеспечивает более высокие показатели производительности и защищенности информации по сравнению с сетевой ОС LAN Manager той же фирмы (совместная разработка с IBM), являющейся надстройкой над локальной операционной системой OS/2.

Виды сетевых ОС

Сетевая служба может быть представлена в ОС либо обеими (клиентской и серверной) частями, либо только одной из них.

В нервом случае операционная система, называемая одноранговой, не только позволяет обращаться к ресурсам других компьютеров, но и предоставляет собственные ресурсы в распоряжение пользователей других компьютеров. Например, если на всех компьютерах сети установлены и клиенты, и серверы файловой службы, то все пользователи сети могут совместно применять файлы друг друга. Компьютеры, совмещающие функции клиента и сервера, называют одноранговыми узлами.

Операционная система, которая преимущественно содержит клиентские части сетевых служб, называется клиентской. Клиентские ОС устанавливаются на компьютеры, обращающиеся с запросами к ресурсам других компьютеров сети. За такими компьютерами, также называемыми клиентскими, работают рядовые пользователи. Обычно клиентские компьютеры относятся к классу относительно простых устройств.

К другому типу операционных систем относится серверная ОС — она ориентирована на обработку запросов из сети к ресурсам своего компьютера и включает в себя в основном серверные части сетевых служб. Компьютер с установленной на нем серверной ОС, занимающийся исключительно обслуживанием запросов других компьютеров, называют выделенным сервером сети. За выделенным сервером, как правило, обычные пользователи не работают.

Примеры сетевых ОС

Сегодня практически все ОС являются сетевыми. Наиболее расропстраненные из них:

- Novell NetWare
- Microsoft Windows (95, NT, XP, Vista, Seven)
- Различные UNIX системы, такие как Solaris, FreeBSD
- Различные GNU/Linux системы

– IOS
– ZyNOS компании ZyXEL
– Chrome OS от Google

Операционные системы мейнфреймов

К высшей категории относятся операционные системы мейнфреймов (больших универсальных машин) — компьютеров, занимающих целые залы и до сих пор еще встречающихся в крупных центрах обработки корпоративных данных. Такие компьютеры отличаются от персональных компьютеров по объемам ввода-вывода данных. Мейнфреймы, имеющие тысячи дисков и петабайты данных — весьма обычное явление, а персональный компьютер с таким арсеналом стал бы предметом зависти. Мейнфреймы также находят применение в качестве мощных веб-серверов, серверов крупных интернет-магазинов и серверов, занимающихся межкорпоративными транзакциями.

Операционные системы мейнфреймов ориентированы преимущественно на одновременную обработку множества заданий, большинство из которых требует колоссальных объемов ввода-вывода данных. Обычно они предлагают три вида обслуживания: пакетную обработку, обработку транзакций и работу в режиме разделения времени. Пакетная обработка — это одна из систем обработки стандартных заданий без участия пользователей. В пакетном режиме осуществляется обработка исков в страховых компаниях или отчетов о продажах сети магазинов. Системы обработки транзакций справляются с большим количеством мелких запросов, к примеру обработками чеков в банках или бронированием авиабилетов. Каждая элементарная операция невелика по объему, но система может справляться с сотнями и тысячами операций в секунду. Работа в режиме разделения времени дает возможность множеству удаленных пользователей одновременно запускать на компьютере свои задания, например запросы к большой базе данных. Все эти функции тесно связаны друг с другом, и зачастую операционные системы универсальных машин выполняют их в комплексе. Примером операционной системы универсальных машин может послужить OS/390, наследница OS/360. Однако эти операционные системы постепенно вытесняются вариантами операционной системы UNIX, например Linux.

Тема 16. Команды ОС Windows тестирования сетевых интерфейсов

Существует несколько инструментов для отслеживания и решения проблем, связанных с применением протокола TCP/IP. Этими инструментами являются PING, ARP, IPCONFIG, TRACERT, NBTSTAT и PATHPING. Все они запускаются из командной строки и выдают результаты в формате DOS. В таблице 8.1 перечислены эти инструменты и дано их краткие описания.

PING

Подобно гидролокатору на подводной лодке, команда PING позволяет получать информацию о своих соседях. Правда, тут она применяется в сугубо мирных целях. Она может сообщить вам о том, как долго информационные пакеты идут из вашего компьютера на принимающий компьютер. Она делает это посредством отправки ICMP эхо-сигнала указанному устройству - будь то устройство локальной сети или сервер на другой стороне земного шара.

Таблица 9 – Инструменты для решения проблем протокола TCP/IP

Инструмент командной строки	**Описание**
ARP	Позволяет модифицировать таблицу протокола разрешения адресов.
IPCONFIG	Показывает текущую TCP/IP конфигурацию и позволяет обновлять эти значения.
NBTSTAT	Предоставляет NetBIOS-информацию о TCP/IP-соединениях, перезагружает кэш LMHost и определяет зарегистрированное имя и область действия ID.
PING	Посылает эхо-запрос на указанное устройство.
TRACERT	Перечисляет количество переходов (изменений маршрута) до указанного устройства.
PATHPING	Показывает степень потери информационных пакетов на любом маршрутизаторе или ссылке.

Если вы тестируете пинг-запросом устройство своей локальной сети, то устройство откликнется практически мгновенно. В этом случае вы узнаете, что оба компьютера работают нормально. При возникновении проблем следует выполнить следующие шаги.

1. Протестируйте пингом-запросом адрес локальной перемычки. Если этот адрес ответит, то на локальном компьютере имеется конфигурация протокола TCP/IP.

Ping 127.0.0.1

2. Протестируйте локальный IP-адрес и убедитесь, что нет конкуренции с другим устройством в сети.

Ping IP_адрес

3. Протестируйте IP-адрес шлюза по умолчанию. Так вы проверите возможность добраться до ближайшего маршрутизатора, который позволяет общаться с компьютерами в другой подсети.

Ping IP_адрес шлюза

4. Протестируйте пингом-запросом адрес указанного вами устройства в другой подсети. Так вы проверите возможность установки связи с устройством другой подсети.

Ping IP_адрес узла

5. Протестируйте пингом-запросом то же самое устройство, применив полное имя его домена. Если попытка закончится провалом, но шаг 4 работает, то это проблема разрешения имени. На этом этапе следует убедиться, что DNS-серверы доступны, таблицы Hosts и LMHosts точны, а WINS (если используется) правильно сконфигурирован.

Ping IP_имя узла

Инструмент PING используется следующим образом:

Ping [-t] [-a] [-n] [-l] [-f] [-I TTL] [-v TOS] [-r] [-s] [-j список узлов] [-k список узлов] [-w] список адресатов

Аргументы PING включают в себя следующее.

- -t Поддерживает пингование, пока не будет остановлен нажатием клавиш CTRL+C.
- -n Посылает эхо-сигнал определенное (указанное) количество раз и прекращает тестирование.
- -l Посылает пакет с указанным количеством битов.
- -f Устанавливает флаг Don't Fragment (Не фрагментировать). Это значит, что пакеты не будут разбиваться на части сетевыми устройствами.
- -w Устанавливает время простоя (мс). Время простоя по умолчанию равно 750 мс.

ARP

Протокол разрешения адресов (Address Resolution Protocol, ARP) позволяет компьютерам создавать соединения на физическом уровне. Независимо от того, используете ли вы NetBIOS или TCP/IP имена компьютеров в своей сети, они должны быть конвертированы в MAC-имена сетевой карты компьютера. Когда одна рабочая станция пытается установить связь с другой, она должна транслировать сигнал в соответствии с протоколом ARP, чтобы выяснить MAC-адрес. После того как Windows XP Professional компьютер определит MAC-адрес, он использует его для установки связи с устройством. Эта конверсия IP в MAC хранится в ARP-таблице компьютера.

Команда ARP позволяет просматривать и редактировать таблицу ARP. Этот инструмент полезен при решении проблем, связанных с разрешениями имен. Команда ARP записывается следующим образом.

ARP -s inet_addr eth_addr [if_addr]

ARP -d inet_addr [if_addr]

ARP -a [inet_addr] [-N if_addr]

В приведенных примерах атрибуты работают следующим образом.

- -s Добавляет IP-адрес (inet_addr) или Ethernet MAC адрес (eth_addr) в таблицу ARP. IP-адрес имеет стандартный четырехоктетный формат, в то время как Ethernet-адрес записывается шестью шестнадцатеричными значениями, разделенными тире.
- -d Удаляет указанный IP-адрес из таблицы.
- -a Выводит на экран текущую ARP-таблицу. Если вы включили в нее IP-адрес, то будет представлена только таблица переводов IP-адреса в MAC-адрес для данного компьютера.

Аргумент [if_addr] указывает IP-адрес, отличный от данного по умолчанию. Если вы хотите посмотреть на таблицу ARP компьютера, которым вы пользуетесь, то введите в командную строку arp-a.

IPCONFIG

Инструмент IPCONFIG хорошо подходит для начала поисков источника проблемы, связанной с применением протокола TCP/IP. Команда записывается следующим образом.

Ipconfig [/all | /release [adapter] | /renew [adapter]]

При использовании без аргументов IPCONFIG представляет только основные настройки TCP/IP, включая IP-адрес, маску подсети и шлюз по умолчанию для каждой карты сетевого адаптера. Однако, добавив аргументы, можно повысить полезность IPCONFIG. Аргументы включают в себя следующее.

- /all Показывает основную и дополнительную информацию, такую как сроки окончания аренды и службы разрешения имен.
- /release Выдает IP-адрес указанному адаптеру, если адаптер использовал DHCP.
- /renew Обновляет IP-адрес для указанного адаптера, если адаптер использовал DHCP.

Примечание. Ввод ipconfig? в командную строку сгенерирует полный список аргументов.

Использование инструмента IPCONFIG может дать огромное количество информации о TCP/IP-соединениях и их конфигурациях. Всегда полезно проверять маску подсети. Убедитесь в том, что она не записана как 0.0.0.0, что указывает на конфликт с другим устройством подсети.

TRACERT

Инструмент Trace Route (TRACERT) применяется для отслеживания перемещения пакета данных от устройства к устройству. Он работает посредством передачи пакета со значением времени жизни (TTL), равным 1. Обычно маршрутизаторы сокращают значение TTL на 1 и затем отправляют пакет дальше по пути следования. Если маршрутизатор получает TTL со значением 0, то он возвращает пакет отправителю как просроченный. Это позволяет узнать кое-что о маршрутизаторе. Инструмент TRACERT выполняет это действие для первого маршрутизатора на пути следования пакета, добавляет 1 к TTL и затем отправляет

новый пакет. Следующий пакет доходит до второго маршрутизатора и становится просроченным. Этот маршрутизатор возвращает пакет вместе с информацией о самом себе. Процесс повторяется, пока пакет не дойдет до нужного устройства, или пока количество переходов не достигнет максимального значения.

Синтаксис команды TRACERT следующий.

Tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout] имя конечного устройства

Некоторые аргументы команды TRACERT описаны ниже.

- -d Препятствует разрешению адреса именам хостов.
- -h maximum_hops Устанавливает верхнюю границу общего числа переходов, необходимых для нахождения нужной рабочей станции.
- -j host-list Устанавливает свободный начальный маршрутизатор для всего списка хостов.
- -w timeout Устанавливает время простоя (мс) для каждого перехода.

Вы можете применять команду TRACERT, просто вводя tracert и адрес конечного устройства.

Этот инструмент полезен, если вы не можете запустить ни одной утилиты из пакета протоколов TCP/IP. После того как вы убедились в том, что TCP/IP установлен, но нельзя использовать команды PING или TRACERT, следует удалить и заново проинсталлировать протокол TCP/IP, который мог повредиться.

NBTSTAT

Инструмент NBTSTAT помогает в решении проблем, связанных с разрешением NetBIOS-имен в TCP/IP-соединениях. Он показывает статистику протокола и текущие TCP/IP-соединения, используя NetBT (NetBIOS поверх TCP/IP). Когда сеть функционирует нормально, NetBT разрешает присваивать NetBIOS-имена IP-адресам.

Команда NBTSTAT имеет следующий синтаксис.

Nbtstat [-a Удаленное имя] [-A IP-адрес] [-c] [-n] [-r] [-R] [-s] [-S] [интервал]

Некоторые аргументы NBTSTAT означают следующее.

- -n Показывает имена, зарегистрированные локально системой, в которой используется сервер или службы переадресации.
- -c Перечисляет переводы имени в IP-адрес, которые находятся в кэше системы.
- -R Заставляет систему очищать кэш и перезагружать его из файла Lmhosts (автоматически перезагружаются только те элементы Lmhosts файла, которые имеют обозначение #PRE).
- -a "имя" Возвращает таблицу NetBIOS-имен компьютера, а также MAC-адрес его сетевой карты.
- -s Перечисляет текущие NetBIOS-сессии, их статус и основные статистические данные.

Примечание. Для получения более подробной информации о NBTSTAT введите nbtstat? в окне команд.

PATHPING

Инструмент PATHPING является комбинацией инструментов PING и TRACERT. Этот инструмент в упорядоченном режиме посылает информационные пакеты на каждый маршрутизатор по пути к месту назначения. Затем он рассчитывает результаты на основании пакетов, возвращенных каждым маршрутизатором. Так как PATHPING показывает степень потери пакетов в любом маршрутизаторе или соединении, администратор может определить, какие именно маршрутизаторы и соединения вызывают проблемы в работе сети.

Команда PATHPING записывается следующим образом.

Pathping [-n] [-h maximum_hops] [-g host-list] [-p period] [-q num_queries] [-w timeout] [-T] [-R] target_name

Некоторые аргументы PATHPING включают в себя следующее.

- -n Не разрешает присваивать адреса именам хостов.
- -h maximum_hops Указывает максимальное количество изменений маршрута, необходимое для нахождения конечного пункта. Настройка по умолчанию предусматривает 30 переходов.
- -p period Указывает время (мс) между двумя передачами пинг-сигнала. По умолчанию равно 250 мс.
- -q num_queries Указывает количество запросов, посланных на каждый компьютер во время прохождения маршрута. Значение по умолчанию - 100.
- -w timeout Указывает время (мс), отводимое на ожидание ответа. По умолчанию - 3000 мс (или 3 с).

Тема 17. Команды ОС Unix конфигурирования и тестирования сетевых интерфейсов

Настройка сетевых интерфейсов

Интерфейсом с точки зрения ОС является устройство, через которое система получает и передает IP-пакеты. Роль интерфейса локальной сети может выполнять одно (или несколько) из следующих устройств: Ethernet-карта, ISDN-адаптер или модем, подключенный к последовательному порту. Каждое устройство (не весь компьютер!) имеет свой IP-адрес. Для выхода в локальные сети используется, как правило, Ethernet-карта, что и будет предполагаться в настоящем разделе.

Расположение конфигурационных файлов

Отметим сразу, что все приводимые ниже команды можно выполнять из командной строки, но тогда придется повторять эти операции при каждом перезапуске компьютера. Поэтому может быть удобнее записать их в один из

инициализационных файлов, автоматически запускаемых при старте системы. В разных дистрибутивах процесс загрузки организован по-разному. В "Linux NET-3-HOWTO" приводится следующая таблица:

Таблица 10 – Расположение конфигурационных файлов в основных дистрибутивах

Дистрибутив	Настройка интерфейса и маршрутизации	Запуск демонов
Debian	/etc/init.d/network	/etc/init.d/netbase /etc/init.d/netstd_init /etc/init.d/netstd_nfs /etc/init.d/netstd_misc
Slackware	/etc/rc.d/rc.inet1	/etc/rc.d/rc.inet2
RedHat	/etc/sysconfig/network-scripts/ifup-<ifname>	/etc/rc.d/init.d/network

Обратите внимание, что дистрибутивы Debian и Red Hat содержат отдельный каталог для скриптов запуска системных сервисов (хотя сами файлы настроек находятся в других местах, например, в дистрибутиве Red Hat они хранятся в каталоге /etc/sysconfig). Для понимания процесса загрузки ознакомьтесь с содержимым файла /etc/inittab и документацией по процессу init.

Команда ifconfig

После подключения драйверов вы должны настроить те интерфейсы, которые вы предполагаете использовать. Настройка интерфейса заключается в присвоении IP-адресов сетевому устройству и установке нужных значений для других параметров сетевого подключения. Наиболее часто для этого используется программа ifconfig (ее название происходит от "interface configuration").

Запустите ее без аргументов (или с единственным аргументом -a) и вы узнаете, какие параметры установлены в данный момент для активных сетевых интерфейсов (в частности, для сетевой карты). Кстати, имеет смысл выполнить эту команду еще до подключения модулей: а вдруг у вас поддержка интерфейсов встроена в ядро и необходимые настройки сделаны в процессе инсталляции системы. Тогда вы в ответ можете получить информацию о параметрах вашей Ethernet-карты и так называемого "кольцевого интерфейса" или "обратной петли" - Local Loopback (интерфейс Ethernet при единственной сетевой карте обозначается как eth0, а кольцевой интерфейс - как lo). Если же по этой команде вы ничего не получите, то надо переходить к подключению модулей и настройке, и начинать надо с кольцевого интерфейса.

Настройка локального интерфейса lo

Этот интерфейс используется для связи программ IP-клиентов с IP-серверами, запущенными на той же машине, так что его необходимо настроить даже в том случае, если вы вообще не подключаете никаких сетевых устройств.

Локальный интерфейс настраивается очень просто: командой

```
[root]# /sbin/ifconfig lo 127.0.0.1
```

Теперь, чтобы проверить работоспособность протоколов TCP/IP на вашей машине, дайте команду:

```
[root]# ping 127.0.0.1
```

Настройка интерфейса платы Ethernet локальной сети (eth0)

Для того чтобы ваш компьютер вошел в сеть с IP-адресом, полученным вами у администратора (пусть для примера это будет адрес 192.168.0.15), вы должны запустить команду ifconfig примерно следующим образом:

```
[root]# /sbin/ifconfig eth0 192.168.0.15 netmask
255.255.255.0 up
```

Если не указывать маску подсети, то по умолчанию устанавливается маска подсети 255.0.0.0.

В некоторых случаях необходимо бывает изменить адрес прерывания, используемого сетевой картой, порта ввода-вывода или типа соединения, используемого в сети. Это можно сделать, выполнив следующую команду:

```
root# /sbin/ifconfig eth0 irq 5 io_addr 220 media
10baseT
```

Не все устройства (платы) поддерживают динамическое изменение этих параметров (т. е. может потребоваться переустановить переключатели на плате).

Интерфейс для последовательного порта

Последовательный порт используется для подключения модема, через который осуществляется соединение с сетью по телефонной линии. Для настройки интерфейса этого типа тоже можно использовать программу ifconfig. Однако, такие программы как pppd и dip, используемые для соединения с сетью по модему, способны автоматически конфигурировать сетевой интерфейс, поэтому обычно для этого случая применять ifconfig не требуется.

Настройка маршрутизации

Правила маршрутизации определяют, куда отправлять IP-пакеты. Данные маршрутизации хранятся в одной из таблиц ядра. Вести таблицы маршрутизации можно статически или динамически. Статический маршрут - это маршрут, который задается явно с помощью команды route. Динамическая маршрутизация выполняется процессом-демоном (routed или gated), который ведет и модифицирует таблицу маршрутизации на основе сообщений от других компьютеров сети. Для выполнения динамической маршрутизации разработаны специальные протоколы: RIP, OSPF, IGRP, EGP, BGP и т. д.

Динамическая маршрутизация необходима в том случае, если у вас сложная, постоянно меняющаяся структура сети и одна и та же машина может быть доступна по различным интерфейсам (например, через разные Ethernet или SLIP интерфейсы). Маршруты, заданные статически, обычно не меняются, даже если используется динамическая маршрутизация.

Для персонального компьютера, подключаемого к локальной сети, в большинстве ситуаций бывает достаточно статической маршрутизации командой route. Прежде чем пытаться настраивать маршруты, просмотрите таблицу маршрутизации ядра с помощью команды netstat -n -r. Вы должны увидеть что-то вроде следующего

[root]# netstat –nr

Kernel IP routing table

Destination	Gateway	Genmask	Flags	MSS	Window	irtt	Iface
10.72.128.101	0.0.0.0	255.255.255.255	UH	0	0	0	eth0
10.72.128.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
127.0.0.0	0.0.0.0	255.0.0.0	U	0	0	0	lo
0.0.0.0	10.72.128.254	0.0.0.0	UG	0	0	0	eth0

Если таблица пуста, то вы увидите только заголовки столбцов. Тогда надо использовать route. С помощью команды route можно добавить или удалить один (за один раз) статический маршрут. Вот ее формат:

```
[root]# /sbin/route [-f] операция [-тип] адресат шлюз
[dev] интерфейс
```

Здесь аргумент операция может принимать одно из двух значений: add (маршрут добавляется) или delete (маршрут удаляется). Аргумент адресат может быть IP-адресом машины, IP-адресом сети или ключевым словом default . Аргумент шлюз - это IP-адрес компьютера, на который следует пересылать пакет (этот компьютер должен иметь прямую связь с вашим компьютером). Команда

```
[root]# /sbin/route -f
```

удаляет из таблицы данные обо всех шлюзах. Необязательный аргумент тип принимает значения net или host . В первом случае в поле адресата указывается адрес сети, а во втором - адрес конкретного компьютера (хоста).

Как правило, бывает необходимо настроить маршрутизацию по упоминавшимся выше трем интерфейсам:

- локальный интерфейс (lo),
- интерфейс для платы Ethetnet (eth0),
- интерфейс для последовательного порта (PPP или SLIP).

Локальный интерфейс поддерживает сеть с IP-номером 127.0.0.1. Поэтому для маршрутизации пакетов с адресом 127.... используется команда:

```
[root]# /sbin/route add -net 127.0.0.1 lo
```

Если у вас для связи с локальной сетью используется одна плата Ethernet, и все машины находятся в этой сети (сетевая маска 255.255.255.0), то для настройки маршрутизации достаточно вызвать:

```
[root]# /sbin/route add -net 192.168.36.0 netmask
255.255.255.0 eth0
```

Если же вы имеете насколько интерфейсов, то вам надо определиться с сетевой маской и вызвать команду route для каждого интерфейса.

Поскольку очень часто IP-пакеты с вашего компьютера могут отправляться не в одну единственную сеть, а в разные сети (например, при просмотре разных сайтов в Интернете), то в принципе надо было бы задать очень много маршрутов. Очевидно, что сделать это было бы очень сложно, точнее просто невозможно. Поэтому решение проблемы маршрутизации пакетов перекладывают на плечи специальных компьютеров - маршрутизаторов, а на обычных компьютерах задают маршрут по умолчанию, который используется для отправки всех пакетов, не указанных явно в таблице маршрутизации. С помощью маршрута по умолчанию вы говорите ядру "а все остальное отправляй туда". Маршрут по умолчанию настраивается следующей командой:

```
[root]# /sbin/route add default gw 192.168.1.1 eth0
```

Опция gw указывает программе route, что следующий аргумент - это IP-адрес или имя маршрутизатора, на который надо отправлять все пакеты, соответствующие этой строке таблицы маршрутизации.

После настройки маршрутизации можно проверить, что у вас получилось. Для этого снова дайте команду

```
[root]# netstat -nr
```

Если вывод команды выглядит так, как это было показано выше, но не содержит строки, которая в графе Destination содержит 0.0.0.0 , а в графе Gateway указывает на маршрут, используемый для соединений по умолчанию, то вы, вероятно, не задали этот маршрут.

Настройка службы имен

С помощью команды ifconfig вы задали IP-адрес вашего компьютера, но он еще не знает своего имени (при инсталляции системы он получил обезличенное имя localhost). Существует команда hostname, которая позволяет установить (и узнать действующее в данный момент) имя компьютера и имя домена.

Однако установить только имя и только этой командой еще недостаточно, поскольку эта команда меняет имя только на текущий сеанс работы. Поэтому обычно эта команда вызывается в одном из инициализационных файлов, например, /etc/rc.d/rc или /etc/rc.d/rc.local. Вы можете попытаться найти ее там, чтобы изменить должным образом имя компьютера, которое задается в качестве параметра команды hostname. В таком случае требуется перезагрузиться для того чтобы изменения вступили в силу.

Другой способ изменения имени компьютера или домена состоит в том, что эти имена прописываются в файле /etc/sysconfig/network в виде двух строчек примерно следующего вида:

```
HOSTNAME="new_host_name.localdomain.upperdomain"
DOMAINNAME=localdomain.upperdomain
```

Тогда в процессе инициализации системы эти имена будут восстанавливаться, потому что файл /etc/sysconfig/network вызывается из /etc/rc.d/rc.sysinit.

Кроме того, имя компьютера должно быть прописано в файле /etc/hosts, который связывает имя компьютера с его IP-адресом. Каждая строка файла /etc/hosts должна начинаться с IP-адреса, за которым следует имя данного узла. Следом за именем можно записать произвольное число псевдонимов этого узла.

Даже если ваш компьютер не подключен к сети, в файле /etc/hosts должна быть прописана хотя бы одна строка следующего вида.

```
127.0.0.1 localhost localhost.localdomain
```

Если же ваш компьютер подключен к TCP/IP сети, то в этом файле дополнительно нужно прописать строку вида

```
192.168.0.15 host_name host_name.localdomain
```

Файл /etc/hosts используется в механизмах разрешения имен. В больших сетях трудно было бы поддерживать в актуальном состоянии файлы /etc/hosts на всех компьютерах, если бы это был основной инструмент для определения IP-адресов по именам. Поэтому обычно для разрешения имен используются серверы DNS. Однако файл /etc/hosts все равно необходим, хотя бы для обращения к серверу DNS. Поэтому в нем имеет смысл указать IP-адреса и соответствующие имена шлюзов и серверов DNS и NIS. А чтобы все приложения использовали этот файл при разрешении имен, должен иметься файл /etc/hosts.conf, содержащий строку

```
order hosts,bind
```

которая говорит, что при разрешении имен сначала должен использоваться файл /etc/hosts, а затем должно происходить обращение к серверу DNS. В большинстве случаев в файле /etc/hosts.conf достаточно иметь две строки:

```
order hosts,bind
multi on
```

Эти параметры указывают системе преобразования имен, что надо просмотреть файл /etc/hosts перед тем, как посылать запрос к серверу, и что следует возвращать все найденные в /etc/hosts адреса для данного имени, а не только первый.

Но настройка механизма разрешения имен не ограничивается редактированием файлов /etc/hosts и /etc/hosts.conf. Необходимо еще указать компьютеру имена серверов DNS. Они прописываются в файле /etc/resolv.conf. Этот файл имеет весьма простой формат. Это текстовый файл, каждая строка которого задает один из параметров системы преобразования имен. Как правило, используются три ключевых слова-параметра:

- domain - задает имя локального домена.
- search - задает список имен доменов, которые будут добавляться к имени машины, если вы не укажете явно имени домена. Это позволяет ограничить область поиска и избежать некоторых ошибок (например, вы ищете компьютер linux.msk.ru, а механизм разрешения имен выведет вас на linux.spb.ru).

- nameserver - этот параметр, который вы можете указывать несколько раз, задает IP-адрес сервера преобразования имен, на который ваша машина будет посылать запросы. Повторяя этот параметр, вы можете задать несколько серверов.

Если вы не собираетесь заводить поддержку сервиса имен для своей сети (что является довольно сложной организационной и технической проблемой), и доверяете ведение своих имен администратору локальной сети или вашему IP-провайдеру, то вам достаточно задать файл /etc/resolv.conf примерно следующего вида:

```
domain abcd.ru
search abcd.ru xyz.edu.ru
nameserver 192.168.10.1
nameserver 192.168.12.1
```

В этом примере машина находится в домене abcd.ru. Если вы зададите имя машины, не указывая домена, например "pc1" , то система преобразования имен попытается сначала найти машину "pc1.abcd.ru" , а в случае неудачи - "pc1.xyz.edu.ru" . Для преобразования имен ваша машина будет обращаться к серверам по адресам "192.168.10.1" и "192.168.12.1".

Тестирование сетевого соединения

Чтобы проверить, соединяется ли ваш компьютер с сетью, попробуйте дать команду ping, указав ей в качестве параметра IP-адрес одного из компьютеров сети. Пусть, например, вам известно (узнайте реальный номер и имя у администратора сети), что в сети есть компьютер с IP-адресом 192.168.0.2 и именем pc1. Тогда вы должны дать команду:

```
[user]$ ping 192.168.0.2
```

или (тут вы одновременно проверяете и работу службы DNS)

```
[user]$ ping pc1
```

Если соединение с сетью установлено, должны появиться и периодически обновляться строчки примерно такого вида:

```
64 bytes from 192.168.0.2: icmp_seq=0 ttl=32 time=1.2 ms
64 bytes from 192.168.0.2: icmp_seq=1 ttl=32 time=1.0 ms
64 bytes from 192.168.0.2: icmp_seq=2 ttl=32 time=1.0 ms
64 bytes from 192.168.0.2: icmp_seq=3 ttl=32 time=1.0 ms
64 bytes from 192.168.0.2: icmp_seq=4 ttl=32 time=1.1 ms
```

Это означает, что сетевое соединение работает. Для того чтобы прервать тестирование сети, нажмите комбинацию клавиш <Ctrl>+<C>.

Программы telnet и rlogin

Для того чтобы воспользоваться программой telnet, вам необходимо знать имя или IP-адрес удаленного компьютера, работающего под управлением ОС типа UNIX, на котором для вас открыт пользовательский бюджет. Предположим для

примера, что на компьютере linux2 имеется пользователь user5, пароль которого вам известен. В таком случае вы можете дать команду

```
[user]$ telnet linux2
```

Если программе удалось подключиться к указанному компьютеру, на экране появится сообщение "Connected to server linux2" и приглашение к входу в систему, как если бы вы сидели за терминалом компьютера linux2. Вводите имя (user5) и пароль, и вы будете работать на этом компьютере.

Команда rlogin может быть использована для выхода на удаленный компьютер вполне аналогично команде telnet, хотя лучше сразу указать в командной строке имя пользователя:

```
[user]$ rlogin -l user5 linux2
```

Завершив работу, не забудьте закрыть сессию (командой exit). После этого программа telnet (или rlogin) докладывает, что сессия закрыта, и вы возвращаетесь к командной строке локальной оболочки.

Когда вы работаете с программой telnet, вы полностью работаете на удаленном компьютере: команды выполняются в его оперативной памяти, вы видите (по команде ls) каталоги и файлы на дисках удаленного компьютера и т. д. Только вывод результатов осуществляется на ваш монитор. В рамках программы telnet невозможно, например, открыть для просмотра файл, расположенный на локальном диске. Ваш компьютер выполняет только роль удаленного терминала. Если же вы хотите организовать обмен файлами между вашим компьютером и удаленным, можно воспользоваться программой ftp.

Программа ftp

Программа ftp - это пользовательский интерфейс к стандартному протоколу передачи файлов по Интернету - File Transfer Protocol. Программа позволяет передавать файлы на удаленный компьютер и получать файлы с удаленного компьютера. Однако, введя команду ftp, вы запускаете только клиентскую программу. Для того чтобы получить доступ к файлам удаленного компьютера, на нем должен быть запущен ftp-сервер. Кроме того, необходимо знать либо имя и пароль пользователя, либо ftp-сервер должен разрешать анонимный доступ. Предположим, что эти условия выполнены и вы запустили программу ftp (без параметров). Вы увидите приглашение интерпретатора команд этой программы:

```
ftp >
```

Если ввести знак вопроса, программа выдаст перечень возможных команд. Первая команда, которую нужно в этом случае ввести, - команда open, после которой надо указать сетевое имя компьютера, на котором запущен ftp-сервер. Если анонимный доступ к этому серверу разрешен, то вы получите запрос на ввод имени и пароля пользователя. По команде pwd можно узнать имя текущего каталога на удаленном компьютере, а по команде dir - вывести список файлов и подкаталогов этого каталога. Команда cd имя_каталога используется для смены текущего каталога на удаленном компьютере.

В любой момент вы можете повторно ввести команду ? или ее эквивалент help , чтобы получить подсказку по возможным командам. Для получения более подробной подсказки по конкретной команде надо ввести имя интересующей вас команды после help или ?, например, так:

```
ftp > help dir
```

Если вы хотите выполнить какую-то команду на локальном компьютере (например, выяснить имя текущего каталога), надо дать соответствующую команду, перед которой поставить восклицательный знак:

```
ftp >! pwd
```

! - это команда интерпретатора, вызывающая новый экземпляр оболочки shell локального компьютера. Первый аргумент, следующий за !, должен быть командой оболочки, а все остальные аргументы - аргументами вызываемой команды. Для смены текущего каталога на локальном компьютере имеется специальная команда lcd (очень полезная, поскольку часто до запуска ftp забываешь перейти в тот каталог, куда хочешь скопировать файл с удаленного компьютера; не выходить же из-за этого из программы ftp).

Для пересылки файла на удаленный компьютер используется команда

```
ftp > put имя_файла
```

(или ее синоним send), а для копирования файла с удаленного компьютера в текущий каталог на локальном диске - команда

```
ftp > get имя_файла
```

В принципе этих двух команд вполне достаточно для организации обмена файлами с удаленным компьютером, но как же неудобно ими пользоваться! Приходится набирать полностью имена всех пересылаемых файлов. Поэтому испытываешь воистину большое облегчение, когда узнаешь, что существуют такие команды как mput и mget. Они позволяют задать шаблон имени пересылаемых файлов, и будут дополнительно переспрашивать, надо ли пересылать каждый конкретный файл. Благодаря этому можно (самый крайний случай) заказать пересылку всех файлов:

```
ftp > mget *
```

а потом либо подтверждать пересылку очередного файла, либо отказываться. Конечно, когда файлов в каталоге очень много, то и это окажется утомительной процедурой, но ведь можно задать более разумный шаблон! Так что думайте, как облегчить себе работу.

Перед тем, как начать пересылку файлов, следует еще выполнить одну из команд, определяющих режим пересылки: ascii или binary . По умолчанию программа использует режим "ascii", и это вполне допустимо при пересылке текстовых файлов, но если вы собираетесь передать или получить исполняемый файл, то необходимо задать режим "binary". Процесс пересылки файлов можно прервать с помощью комбинации клавиш <Ctrl>+<C>.

Пока вы находитесь в программе ftp, вы можете выполнить некоторые операции с файлами и каталогами на удаленном компьютере (конечно, для этого надо иметь соответствующие права). По команде

```
ftp > rename from_name to_name
```

осуществляется переименование файла или каталога; команда

```
ftp > mkdir name
```

создает каталог, а

```
ftp > delete name
```

удаляет файл или каталог. Еще одна интересная команда - system, позволяет выяснить тип операционной системы на удаленном компьютере. Ну, и наконец, команда close (или disconnect) позволяет завершить сеанс работы с удаленным компьютером, не выходя из программы ftp (т. е. предполагается, что после этого вы снова дадите команду open, например, для соединения с другим компьютером). Если же вы хотите вообще выйти из программы, то надо дать команду bye.

Виртуальные терминалы и интерфейс командной строки NetWork Simulator'a.

Виртуальные устройства в NET-Simulator управляются при помощи интерфейса командной строки из виртуальных терминалов. Терминал устройства можно открыть двойным кликом на значке устройства или через контекстное меню. Поддерживается история команд, клавиши вверх/вниз позволяют просматривать историю команд.

Список команд доступных на данном устройстве можно посмотреть командой help. Сочетание клавиш Ctrl+L очищает терминал. Краткая справка по любой команде выводится при вызове команды с опцией -h.

Справочник команд:

- help
- route
- ifconfig
- ping
- arp
- mactable

help — выводит список доступных команд.

help [-h]Опции Описание

-h Краткая справка.

route — позволяет управлять таблицей маршрутизации устройств поддерживающих протокол IP4.

route [-h] [{-add|-del} <target> [-netmask <address>] [-gw <address>] [-metric <M>] [-dev <If>]]Опции Описание

-h Краткая справка.

target Адрес назначения. Назначением может быть подсеть или отдельный узел в зависимости от значения маски подсети. Если маска равна 255.255.255.255 или отсутствует совсем назначением будет узел, иначе назначением будет сеть.

-add Добавляет новый маршрут в таблицу маршрутизации.

-del Удаляет маршрут из таблицы маршрутизации.

-dev <If> Принудительно присоединяет маршрут к определенному интерфейсу. If — имя интерфейса.

-gw <address> Направляет пакеты по этому маршруту через заданный шлюз. address — адрес шлюза.

-netmask <address> Маска подсети используемая совместно с адресом назначения при добавлении маршрута. address — маска. Если маска не задана явно подразумевается 255.255.255.255.

-metric <M> Метрика используемая в данном маршруте. M — целое число большее или равное нулю.

Если route вызывается без параметров, то команда выводит на экран таблицу маршрутизации:

```
=>route
IP routing table
Destination     Gateway     Netmask         Flags   Metric  Iface
10.0.0.0        *           255.0.0.0       U       1       eth0
11.0.0.0        10.0.0.10   255.0.0.0       UG      1       eth0
192.168.120.1   10.0.0.10   255.255.255.255 UGH     1       eth0
```

Если маршрут не использует шлюз, вместо адреса шлюза выводиться *. Flags может содержать значение: U — маршрут активен, G — маршрут использует шлюз, H — назначением является узел.

Примеры:

```
=>route -add 192.168.120.0 -netmask 255.255.255.0 -dev eth0
=>route
IP routing table
Destination     Gateway         Netmask         Flags   Metric  Iface
192.168.120.0   *               255.255.255.0   U       1       eth0
=>
=>route -add 192.168.121.10 -gw 192.168.120.10
=>route
```

```
IP routing table
Destination     Gateway         Netmask         Flags Metric Iface
192.168.120.0   *               255.255.255.0   U     1      eth0
192.168.121.10  192.168.120.1   255.255.255.255 UGH   1      eth0
=>
```

ifconfig — конфигурирует сетевые интерфейсы.

ifconfig [-h] [-a] [<interface>] [<address>] [-broadcast <address>] [-netmask <address>] [-up|-down] Опции Описание

-h Краткая справка.

-a Показывать информацию о всех интерфейсах. Если данная опция отсутствует выводится информация только об активных интерфейсах.

interface Конфигурировать или показать информацию только о заданном интерфейсе.

address IP-адрес присваиваемый интерфейсу.

-broadcast <address> Широковещательный адрес присваиваемый интерфейсу. address — широковещательный адрес.

-netmask <address> Маска подсети используемая совместно с адресом. address — маска. Если маска не задана явно, маска принимается равной стандартным значения для стандартных классов подсетей A, B и C.

-up Активирует интерфейс. При активизации интерфейса для него автоматически добавляется соответствующий маршрут в таблице маршрутизации.

-down Деактивирует интерфейс. При деактивации интерфейса соответствующий маршрут автоматически удаляется из таблицы маршрутизации.

Если ifconfig вызывается без параметров, то команда выводит на экран данные о состоянии всех активных интерфейсов:

```
=>ifconfig
eth0   Link encap:Ethernet  HWaddr 0:0:0:0:CF:0
       inet addr:192.168.120.1  Bcast:192.168.120.255  Mask:255.255.255.0
       UP
       RX packets:23 errors:0 dropped:0
       TX packets:23 errors:0 dropped:0
       RX bytes:0 TX bytes:0
```

HWaddr — уникальный 6-ти байтовый адрес интерфейса, аналогичный MAC-адресу в Ethernet сетях. Назначается автоматически.

Примеры:

```
=>ifconfig eth0 192.168.120.1 -up
```

```
=>ifconfig
eth0    Link encap:Ethernet  HWaddr 0:0:0:0:CF:0
        inet addr:192.168.120.1  Bcast:192.168.120.255  Mask:255.255.255.0
        UP
        RX packets:0 errors:0 dropped:0
        TX packets:0 errors:0 dropped:0
        RX bytes:0 TX bytes:0
```

ping — использует ICMP протокол что бы проверить достижимость интерфейса удаленного узла. ping посылает удаленному узлу ICMP ECHO_REQUEST и ожидает в течении определенного промежутка времени ICMP ECHO_RESPONSE. В случае получения ответа выводит данные о прохождении ICMP-пакета по сети.

ping [-h] [-i <interval>] [-t <ttl>] <destination> Опции Описание

-h Краткая справка.

-i <interval>Задает частоту ICMP-запросов. interval — интервал между запросами в секундах. По умолчанию отсылается один пакет в секунду.

-t <ttl> Задает значение атрибута Time to Live в генерируемых IP-пакетах. ttl — целое число 0-255. По умолчанию TTL равно 64.

destination IP-адрес исследуемого узла

Примеры:

```
=>ping 192.168.120.1
PING 192.168.120.1
64 bytes from 192.168.120.1: icmp_seq=0 ttl=62 time=477 ms
64 bytes from 192.168.120.1: icmp_seq=1 ttl=62 time=435 ms
64 bytes from 192.168.120.1: icmp_seq=2 ttl=62 time=234 ms
64 bytes from 192.168.120.1: icmp_seq=3 ttl=62 time=48 ms
64 bytes from 192.168.120.1: icmp_seq=4 ttl=62 time=87 ms
64 bytes from 192.168.120.1: icmp_seq=5 ttl=62 time=56 ms
```

ping выводит результат исследования удаленного узла в следующем формате: 64 bytes from 192.168.120.1 — размер полученного ответа и адрес источника ответа. В NET-Simulator размер пакета имеет условное значение и всегда равен 64B. icmp_seq=0 — номер пакета. Каждый запрос содержит свой номер, как правило формируется инкрементно. ping выводит номер пакета из каждого полученного ответа. ttl=62 — значение TTL из полученного ответа. time=48 ms — время прохождения пакетом полного маршрута (туда и обратно, round-trip time) в миллисекундах.

arp — показывает ARP-таблицу устройства. Кроме того опция -r позволяет сформировать запрос для определения MAC-адреса по явно заданному IP-адресу. Эта функция обычно отсутствует в реальных устройствах, в NET-Simulator она добавлена для наглядности при изучении протоколов канального и сетевого уровня.

arp [-h] [-r <IP-address> <interface>]Опции Описание

-h Краткая справка.

-r <IP-address> <interface> Прежде чем вывести ARP-таблицу предпринимает попытку найти MAC-адрес по явно заданному IP-адресу. IP-address IP-адрес для которого определяется MAC-адрес. interface имя интерфейса в сети подсоединенной к которому будет происходить поиск.

Если arp вызывается без параметров, то команда выводит на экран ARP-таблицу:

```
=>arp
Address         HWaddress       iface
10.0.0.10       0:0:0:0:BC:0    eth0
10.0.0.11       0:0:0:0:1F:2    eth0
```

Примеры:

```
 =>arp -r 192.168.120.12 eth1
Address         HWaddress       iface
10.0.0.10       0:0:0:0:BC:0    eth0
10.0.0.11       0:0:0:0:1F:2    eth0
192.168.120.12   0:0:0:0:12:1    eth1
```

mactable — показывает таблицу MAC-адресов коммутаторов второго уровня.

mactable [-h]Опции Описание

-h Краткая справка.

Примеры:

```
=>mactable
MACAddress      port
0:0:0:0:B3:0    0
0:0:0:0:2F:2    0
0:0:0:0:03:0    3
```

где port — номер порта на коммутаторе. Нумерация портов идет по порядку начиная с нуля.

РАЗДЕЛ 6. ГЛОБАЛЬНЫЕ СЕТИ

Тема 18. Основные принципы построения глобальных сетей

Структура глобальных сетей

Глобальные сети *(WAN, Wide Area Networks) позволяют организовать взаимодействие между компьютерами на больших расстояниях*. В идеале глобальная компьютерная сеть должна передавать данные абонентов любых типов, которые есть на предприятии и нуждаются в удаленном обмене информацией. Для этого глобальная сеть должна предоставлять целый комплекс услуг: передачу пакетов локальных сетей, обмен факсами, передачу телефонных разговоров, обмен видеоизображениями и т.д.

Из приведенного выше перечня услуг глобальных сетей видно, что они используются в основном как транзитный транспортный механизм, предоставляющий только услуги трех нижних уровней модели OSI. В последнее время, однако, в связи с развитием сети Internet, где представлен самый широкий спектр услуг протоколов верхнего уровня (WWW, News и д.р.), к сетевым ресурсам глобальных сетей предъявляет новые требования. В Наблюдается сближения технологий глобальных и локальных сетей на разных уровнях модели OSI - от транспортных до прикладных.

В общем случае глобальная сеть строится с помощью каналов связи, которые соединяются коммутаторами глобальной сети. Такие коммутаторы называются также *центрами коммутации пакетов (ЦАП)*. Отметим, что в зависимости от технологий передачи данных в глобальных сетях пакеты могут называться *кадры, ячейки*. Коммутаторы устанавливаются в тех географических пунктах, в которых требуется ответвление или слияние потоков данных конечных абонентов или магистральных каналов, переносящих данные многих абонентов. Выбор места установки коммутаторов определяется многими факторами: наличием квалифицированного персонала по обслуживанию в данной местности, надежностью местной сети, наличием необходимых линий абонентов, стоимостью и т.д. Абоненты сети подключаются к коммутаторам также по *выделенным каналам связи*, которые имееют более низкую пропускную способность, чем *магистральные каналы*. Для подключения конечных пользователей допускается использование *коммутируемых* (не выделенных) каналов, т.е. каналов телефонных сетей. Такие каналы имеют значительно низшее качество связи из- высокого уровня шумов. Конечные узлы глобальной сети более разнообразны, чем конечные узлы локальной сети. На рис 1. показан пример построения глобальной сети. Как видно из рисукна к глобальной сети могут подключаться как отдельные домашние компьютеры, так и целые локальные сети, телефонные станции, а также друигие устройства,

требующие каналов связи. При этом для совмещения передачи компьютерных и голосовых данных используются специальные устройсва, называемые *мультиплесорами*. Обычно предача голоса имеет более высокий приоритет.

При передаче данных через глобальную сеть маршрутизаторы и коммутаторы работают с той же логикой, что и в локальных сетях. В этом случае последние называются *удаленными коммутаторами*. Маршрутизаторы принимают решение о пересылке пакетов на основании номера сети какого либо протокола сетевого уровня (например IP) . Следует при использовании предприятием глобальной сети четко выяснить перечень предоставляемых глобальной сетью услуг, а также определить интерфейс взаимодействия сети предприятия с глобальной сетью, чтобы его оборудование и программное обеспечение корректно сопрягалось с соответствующим оборудованием и программным обеспечением глобальной сети. Протоколы взаимодействия глобальной сети с абонентами называются *интерфейсы пользователь- сеть (User-to-Network Interface, UNI)*, а протоколы взаимодействия коммутаторов внутри глобальной сети называются *интерфейсами сеть – сеть (Network- to Network Interface, NNI*). Аппаратура (устройства), вырабатывающие данные для передачи в глобальную сеть называются устройствами *DTE (Data Terminal Equipment)*- порт маршрутизатора, модем домашнего пользователя и т.д. Так как с этих устройств данные передаются в глобальную сеть по каналу связи, имеющему определенный стандарт, то устройства DTE оснащаются устройствами, называемыми DCE (Data Circuit terminating Equipment). Между этими устройствами существует свой интерфейс (стандарт), наиболее популярный из известных – RS-232 C/ V245. Он представляет собой 25 контактный разъем, где на каждый контакт должен поступать в соответствии со стандартом определенный сигнал (питание, земля, передающий сигнал, принимаемый сигнал, сигнал синхронизации и т.д.). Скорость передачи данных до 115 200 бит/ с. Указанный интерфейс, например, реализован во всех компьютерах в виде COM – порта.

Кроме этого существуют другие интерфейсы – RS 449, V35, X 21, HSSI (High – Speed Serial Interface) Некоторые из них могут поддерживать скорость до 10 Мбит на расстоянии до 10 метров.

Типы глобальных сетей

При построения глобальной сети необходимо учитывать множество различных факторов, но главными из них является выбор типа глобальной сети, или, другими словами выбор метода организации каналов передачи информации.

В общем случае выделяют три типа глобальных сетей. Это сети с использованием:

- выделенных каналов
- коммутации каналов
- коммутации пакетов

Отметим особенности каждого из типов глобальных сетей.

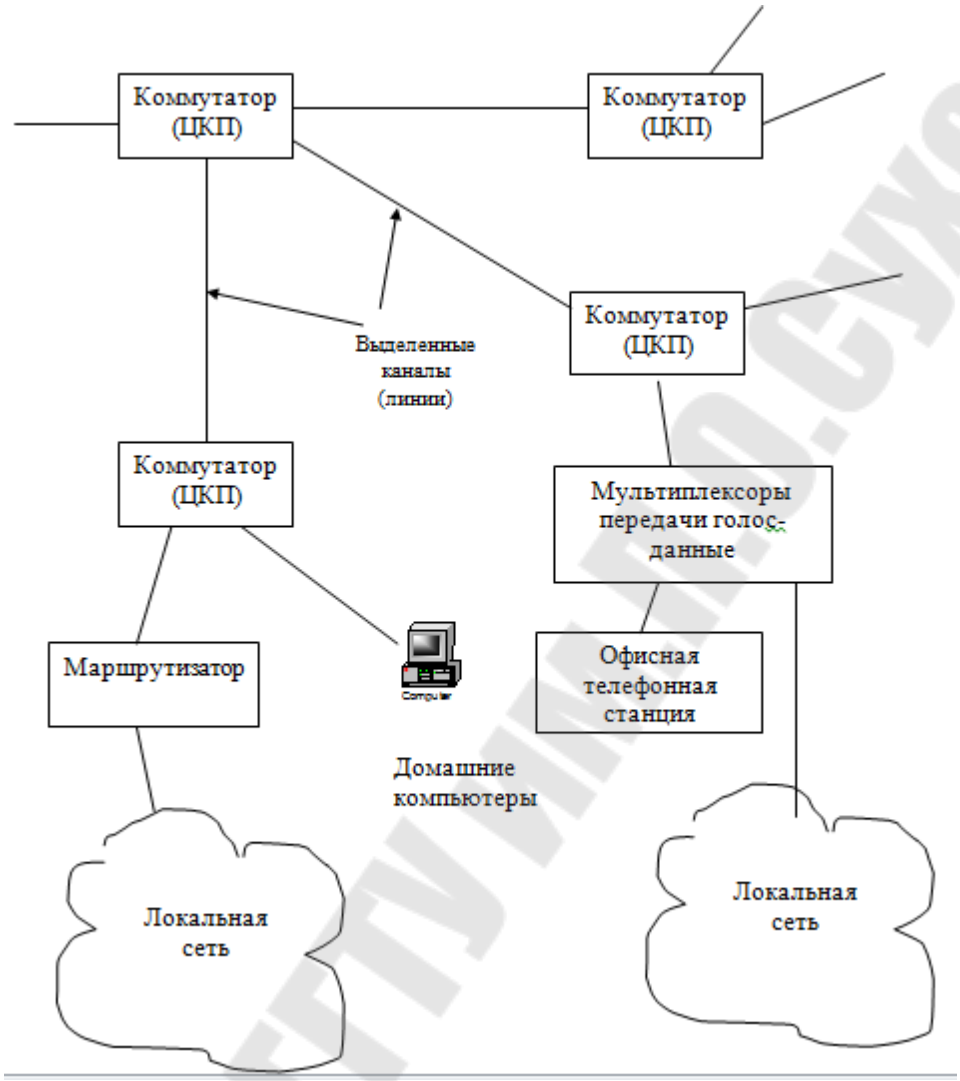


Рисунок 49 – Структура глобальной компьютерной сети

Выделенные каналы

Выделенные каналы можно получить у телекоммуникационных компаний (в Республике Беларусь, например, у Белтелеком), которые владеют каналами дальней связи и сдают их в аренду.

Использовать выделенные линии можно двумя способами:

- территориальная сеть строится с их помощью для соединения между собой коммутаторов (как на рис. 49);
- соединение между собой только объединяемых локальных сетей или конечных абонентов другого типа.

Второй случай является более простым и предпочтительным, т.к. так как отсутствуют протоколы глобальных сетей. Иногда второй способ называется «услуги выделенных каналов», так как в нем действительно больше ничего не используется из технологий собственно глобальных сетей. Выделенные каналы

активно применяются сегодня для связи между крупными локальными сетями, так эта услуга гарантирует пропускную способность арендуемого канала. При большом, однако, количестве географически удаленных точек и интенсивном трафике использование выделенных каналов приводит к высоким затратам за счет большого числа арендуемых каналов.

Глобальные сети с коммутацией каналов

Сети с коммутацией каналов в настоящее время используют каналы двух типов: традиционные аналоговые телефонные каналы и цифровые каналы с интеграцией услуг ISDN (будет рассмотрена в следующей лекции). Преимуществом сетей с коммутацией каналов является их широчайшая распространенность – обычные телефонные сети. Последнее время сети ISDN также стали использоваться в нашей республике.

Основным недостатком аналоговых телефонных сетей является низкое качество составного канала из-за перекрестных частотных помех Цифровые каналы связи лишены указанных недостатков, так как по каналу передается сигналы в специальном цифровом кодировании. Сети с коммутацией каналов имеют тот недостаток, что пользователь платит не за объем переданной или полученной информации, а за время подключения. Одна для работы дома телефонные каналы связи являются единственной возможностью выхода в глобальную компьютерную сеть.

Глобальные сети с коммутацией пакетов.

Принцип коммутации пакетов был рассмотрен в предыдущих лекциях. К таким сетям относятся в настоящее время такие технологии как X25, frame relay, SDMS и ATM, которые будут подробнее рассмотрены ниже.

Магистральные сети и сети доступа

Территориальные глобальные сети можно разделить на две большие категории:

- Магистральные сети
- Сети доступа

Магистральные сети используются для образования связей между крупными локальными сетями, принадлежащим большим подразделениям. Они должны обеспечить высокую пропускную способность, т.к. на магистрали объединяются потоки большого количества сетей. Кроме этого они должны обеспечивать высокий коэффициент готовности, т.к. через них может проходить очень важная оперативная информация. Обычно в качестве магистральных сетей используются цифровые выделенные каналы со скоростями от 2 до 622 Мбит / с и используются технологии сетей frame relay, ATM, X25 или TCP/ IP сети. Для обеспечения высокой готовности магистрали используется смешанная избыточная топология.

Под сетями доступа понимаются территориальные сети, необходимые для связи небольших локальных сетей и отдельных удаленных компьютеров с

центральной сетью предприятия. Вопросам удаленного доступа в последнее время уделяется особенно важное значения, т.к. быстрый доступ к корпоративной информации из любой географической точки является в настоящее время важным фактором для своевременного принятия управленческих решений. В качестве удаленных узлов могут быть также банкоматы или кассовые аппараты. К сетям доступа предъявляются требования, существенно отличающиеся от требований к магистральным сетям. Так как точек удаленного доступа может быть много, то в этом случае сеть доступа должна иметь очень разветвленную структуру, которая может быть использована сотрудниками как дома, так и в командировках. Кроме этого стоимость удаленного доступа должна быть не высокой, чтобы экономически оправдать большое число удаленных пользователей. В качестве сетей доступа обычно применяют телефонные аналоговые сети, сети ISDN, иногда сети frame relay. Для таких сетей используются каналы со скоростью 64 Кбит/с – 2 Мбит/с.

Программные и аппаратные средства , которые обеспечивают подключение компьютеров или локальных сетей удаленных пользователей к глобальной сети называются *средствами удаленного доступа* . Обычно на клиентской стороне это модем и соответствующее программное обеспечение.

Организацию массового удаленного доступа со стороны сети обеспечивает обычно *сервер удаленного доступа (Remote Access Server, RAS)*. Такие сервера имеют много низкоскоростных портов для подключения пользователей через аналоговые телефонные сети или ISDN.

На рис. 50 показана структура глобальной сети, объединяющая в корпоративную сеть отдельные локальные сети и удаленных пользователей.

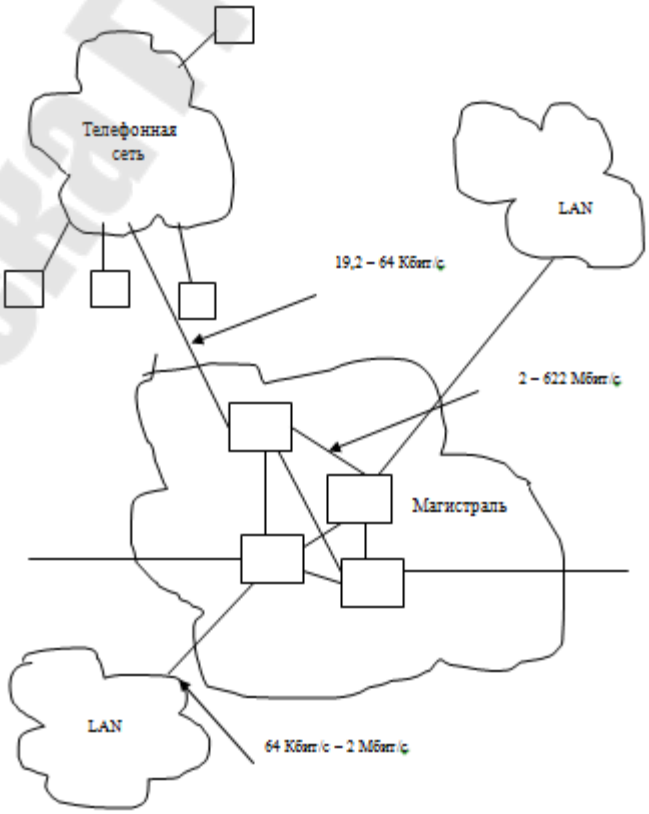


Рисунок 50 – Объединение локальных сетей в глобальную сеть

Глобальные сети на основе выделенных каналов

Выделенный канал – это канал с фиксированной полосой пропускания или фиксированной пропускной способностью, постоянно соединяющий двух абонентов. Абонентами могут быть как отдельные устройства (компьютеры или терминалы), так целые сети.

Выделенные каналы делятся на *аналоговые и цифровые*..

Аналоговые выделенные каналы**.**

Аналоговые выделенные каналы (линии) могут быть 2-х проводные или 4-проводные. В 4- х проводных линиях два провода используются на прием, а два на передачу, что значительно увеличивает пропускную способность канала.

Аналоговые выделенные каналы делятся на *нагруженные и ненагруженные*.

Первую группу составляют линии , проходящие через аппаратуру телефонных станций, и работают на тональных частотах от 3,1 КГц до 108 КГц.

Вторая группа выделенных аналоговых линий – это линии, которые не проходят через аппаратуру уплотнения телефонных станций. Такие линии обладают широкой полосой пропускания (до 1 МГц).

Для передачи информации по аналоговым линиям связи используется частотное разделение каналов, где каждый канал имеет собственную частотную полосу. Поэтому недостатком таких линий связи является влияние каналов друг на друга, т.е. наличие перекрестных частотных помех.

Для передачи данных по выделенным нагруженным аналоговым линиям связи используются специальные устройства, называемые *модемами.* Модемы преобразуют цифровой сигнал в аналоговый с помощью методов аналоговой модуляции. В зависимости от режимов работы различные модемы обеспечивают различные скорости передачи данных : от 1200 бит/с до 33,6 Кбит/с.

Цифровые выделенные каналы

Цифровые выделенные линии представляют собой постоянные линии связи, работающие на принципе разделения каналов по времени. Исторически существует для таких линий две технологии передачи данных: более ранняя PDH (Plesiochronic Digital Hierarchy) – почти синхронная (плезио) иерархия и SDH (Synchronous Digital Hierarchy) – синхронная иерархия. В США синхронная иерархия реализована в стандарте SONET.

Основной принцип технологии *PDH* заключается в объединении на более высоком уровне в один канал несколько более низкоскоростных каналов. В начале (60-е годы 20-го столетия) была разработана аппаратура низшего уровня (ее назвали

аппаратура Т1), которая объединила в цифровом виде на постоянной основе 24 абонента. Каждый абонентский канал образовывал поток данных скоростью 64 Кбит/ с. Затем на более высоком уровне четыре канала Т1 объединялись в более скоростной канал Т2, передающий данные со скоростью 6,312 Мбит/с. В свою очередь семь каналов Т2 объединялись в канал Т3, имеющий скорость 44,736 Мбит/с. АппаратураТ1,Т2,Т3 образует при взаимодействии иерархическую сеть (Рис.51).

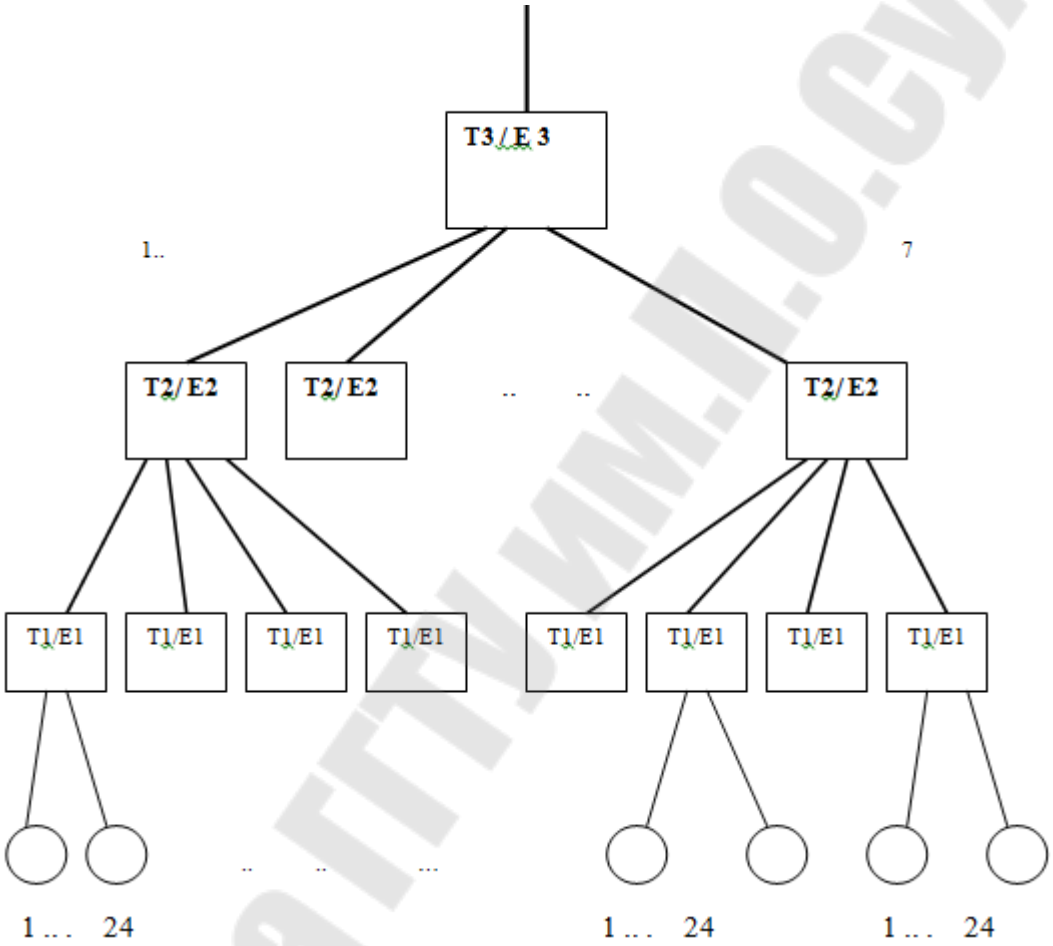


Рисунок 51 – Иерархическая сеть PDH

Глобальные сети с коммутацией каналов

Глобальные сети с коммутацией каналов используют услуги телефонных сетей. Телефонные сети делятся на *аналоговые и цифровые* в зависимость от способов коммутации (мультиплексирования) абонентских и магистральных каналов. Аналоговые телефонные сети принимают данные от абонентов в аналоговой форме, а мультиплексирование и коммутацию осуществляют как аналоговым, так и цифровым методами. В цифровых сетях информация от абонентов поступает в цифровом виде, и используются цифровые методы коммутации.

Аналоговые телефонные сети

Наиболее популярными аналоговыми коммутируемыми каналами являются обычные телефонные сети. Такие сети в настоящее время малопригодны для

построения магистралей, так как их максимальная пропускная способность составляет 56 Кбит/с, да и то в случае использования цифровых коммутаторов. В общем случае средняя пропускная способность аналоговых телефонных сетей составляет 9600 бит/с. В настоящее время аналоговые телефонные сети используются для организации индивидуального удаленного доступа, например, подключения с домашнего компьютера к сети Интернет. Соединение локальных сетей с помощью аналоговых коммутируемых каналов является экономически невыгодным из- за низкой пропускной способности и необходимости оплачивать не количество передаваемой информации, а время соединения. Обычно такие каналы для соединения локальных сетей рекомендуется использовать только для передачи сводок, имеющих небольшие объемы. Для подключения к физическим линиям связи используются как модемы, используемые только для коммутируемых каналов, так и модемы универсальные, применяемые также и на выделенных каналах. Последние стоят дороже. В отличие от модемов для выделенных каналов в модемах для коммутируемых каналов существует функция набора телефонного номера.

Цифровые телефонные сети

К первым цифровым телефонным сетям относятся так называемые службы Switched 56 (коммутируемые каналы 56 Кбит/с) и цифровые сети с интегральными услугами ISDN (Intergrated Services Digital Network). В настоящее время в мире наблюдается тенденция вытеснения службы Switched 56 сетями ISDN.

Сети ISDN

В сетях ISDN данные обрабатываются в цифровом виде. Первоначально сети создавались для передачи голоса, но они могут также использоваться и для передачи компьютерных данных.

В сетях ISDN используются (интегрируются) несколько видов служб: *выделенные цифровые канал; коммутируемая телефонная сеть общего пользования:, сеть передачи данных с коммутацией пакетов, сеть передачи данных с трансляцией кадров (frame relay); средства контроля и управления сетью*. Стандарты ISDN описывают также ряд других услуг прикладного уровня: факсимильную связь на скорости 64 Кбит/ с, телексную связь на скорости 9600 Бит/с, видеотелекс на скорости 9600 Бит/с и некоторые другие. Базовой скоростью сети является скорость 64 Кбит/ с. Сеть поддерживает два типа пользовательского интерфейса: начальный интерфейс BRI (Basic Interface Interface) и основной интерфейс (Primay Rate Interface, PRI).

Начальный интерфейс BRI предоставляет пользователю два канала по 64 Кбит/с для передачи данных (каналы типа B) и один канал с пропускной способностью 16 Кбит/с для передачи управляющей информации (канал типа D). Все каналы работают в полнодуплексном режиме, что позволяет получить суммарную скорость передачи данных 144 Кбит/с.

Основной интерфейс PRI предназначен для пользователей с повышенными требованиями к пропускной способности. Интерфейс PRI поддерживает либо схему каналов 30 B+ D, либо схему 23 B + D. В обеих схемах канал D обеспечивает скорость 64 Кбит/ с. Первый вариант предназначен для Европы, второй для Северной Америки и Японии, соответствующими скоростями 2,0248 Мбит/с и 1,544 Мбит/с.

В настоящее время сети ISDN используются в основном как телефонные цифровые сети высокого качества. Их преимущество, по сравнению с традиционными телефонными сетями, является то, что они позволяют организовать одновременно несколько цифровых каналов через один телефонный провод и объединить различные транспортные и прикладные службы. В качестве магистралей указанные сети не используются из- за отсутствия скоростной службы коммутации пакетов и невысокие скорости каналов.

В сетях с коммутацией каналов в основном используются пакеты небольшого фиксированного размера.

Тема 19. Глобальные сети с коммутацией пакетов

Для глобальных сетей с выделенными или коммутируемыми каналами основные проблемы были сосредоточены на физическом и канальном уровне, так как на сетевом уровне работали сетевые протоколы IP или IPX, с помощью которых происходило объединение локальных сетей.

Для глобальных сетей с коммутацией пакетов используется другая оригинальная техника маршрутизация пакетов, использующая понятие *«виртуального канала»*.

Техника виртуальных каналов используется во всех сетях с коммутацией пакетов, кроме сетей TCP/IP.

Виртуальный канал устанавливается между абонентами сети перед тем, как начать передачу пакета с помощью посылки в сеть специального пакета – запрос на установление соединения (Call Request), который содержит адрес узла назначения.

Существуют два типа виртуального канала*: коммутируемый виртуальный канал (SVC- Switched Virtual Circuit) и постоянный виртуальный канал (Permanent Virtual Circuit)*.

При создании *коммутируемого виртуального канала* коммутаторы сети настраиваются по запросу абонента (динамически), а создание постоянного виртуального канала происходит заранее, причем коммутаторы сети настраиваются вручную администратором сети. Принцип работы виртуального канала состоит в том, что маршрутизация пакетов между коммутаторами сети на основании таблиц маршрутизации происходит только *один раз* – при создании виртуального канала. После создания виртуального канала пакеты предаются с помощью так называемых

номеров или идентификаторов виртуальных каналов (VCI- Virtual Channel Identifier).

После прокладки виртуального канала через глобальную сеть коммутаторы больше не используют для пакетов этого соединения таблицу маршрутизации, а продвигают пакеты на основании номеров виртуальных каналов. Поскольку таблицы коммутации портов значительно меньше таблиц маршрутизации (в них содержаться только текущие соединения), то и пересылка пакетов осуществляется с большей скоростью.

Режим постоянного виртуального канала (PVC) является особенностью технологии маршрутизации в глобальных сетях. Отметим, что в сетях TCP/IP такого режима нет. Режим PVC является наиболее эффективным с точки зрения производительности сети, так большую часть работ по маршрутизации пакетов администратор сети выполняет на этапе подготовки.

Техника виртуальных каналов имеет свои достоинства и недостатки по сравнению с техникой IP- маршрутизации. Маршрутизация пакетов без предварительного установления соединения (IP- адресация) эффективна для кратковременных потоков данных. Кроме этого, если имеются дополнительные параллельные линии связи, пакеты могут продвигаться по ним, что увеличивает надежность сети. При использовании же виртуальных каналов очень эффективно передаются долговременные потоки, так как для установления виртуального канала требуется дополнительное время (5- 10 мс), что для кратковременных пакетов является неприемлемым.

Рассмотрим наиболее популярные сети с коммутацией пакетов.

Сети X.25

Технология сетей X.25 - самая старая из стандартных технологий построения территориальных сетей с коммутацией пакетов. До использования Интернет в коммерческих целях сети X.25 были единственными доступными для коммерческих целей сетями. Сети X.25 хорошо работают на ненадежных и зашумленных линиях связи за счет установления виртуального соединения и коррекции ошибок на двух уровнях- канальном и сетевом. Стандарт X.25 был разработан в 1974 году. Сети X.25 хорошо подходят для передачи трафика низкой интенсивности, например, для подключения удаленного терминала (например, банкомат, касса). Среди особенностей сети X.25 является то, что она может работать только с одним протоколом канального уровня, который называется LAB- B, т.е. в отличие от IP-сетей не может объединять разнородные локальные сети.

Сеть X.25 состоит из коммутаторов, соединенных высокоскоростными выделенными каналами.

Для адресации сетей X.25 и их соединения между собой используется международная нумерация, называемая IDN – International Data Numbers. Адреса имеют разную длину, которая может доходить до 14 десятичных знаков. Первые

четыре цифры IDN называются кодом идентификации сети (DNIC- Data Network Identification Code). DNIC поделен на две части: первая (три цифры) определяют страну нахождения сети, а вторая номер сети X.25 внутри страны. Для нумерации сети внутри страны остается только одна цифра, что позволяет иметь внутри страны только 10 сетей X.25. Если в стране требуется более чем 10 сетей X.25, то стране присваивается дополнительный номер. Остальные цифры используются для адресации пользователей внутри сети.

Коммутаторы сетей X.25 представляют собой более простые и дешевые устройства, чем маршрутизаторы сетей TCP/IP. Это связано с тем, что коммутаторы не выполняют операций преобразований различных форматов канального уровня (как IP- маршрутизатор для объединения локальных сетей различных технологий) и не определяют оптимальный путь прохождения пакетов. В отличие от коммутаторов локальных сетей коммутаторы сетей X.25 обмениваются информацией подтверждения о получении кадров и организуют повторную передачу утерянного кадра.

Отметим, что сети X.25 были разработаны для низкоскоростных линий связи (1200 – 9600 бит/с) с высоким уровнем помех, которые еще широко распространены в нашей республике.

Сети frame relay

Технология framerelay начинает занимать в территориальных сетях с коммутацией пакетов ту же нишу, которую заняла в локальных сетях технология Ethernet. Обе технологии предоставляют только базовый транспортный сервис, доставляя кадры в узел назначения без гарантий, дейтаграммным способом. Однако, если кадры теряются, то сеть framerelay, как и сеть Ethernet, не предпринимает никаких усилий для их восстановления. Поэтому полезная пропускная способность сервисов верхнего уровня в сетях framerelay зависит от качества каналов и методов восстановления пакетов протоколами верхнего уровня, расположенными над протоколом framerelay. Если каналы качественные, то кадры будут теряться и искажаться редко, так что скорость восстановления пакетов протоколом TCP или NCP будет вполне приемлема. Если же кадры искажаются и теряются часто, то полезная пропускная способность в сети framerelay может упасть в десятки раз, так, как это происходит в сетях Ethernet при плохом состоянии кабельной системы.

Поэтому сети framerelay неразрывно связаны с оптоволоконными кабелями, по крайней мере на магистральных каналах "коммутатор - коммутатор".

На оптоволоконных линиях связи они обеспечивают передачу данных со скоростью до 2 Мбит/с. Технология Frame relay использует для передачи данных технику виртуальных соединений, аналогичную техники сетей X.25.

В отличие от сетей X.25 и сетей TCP/ IP, однако, в сетях frame relay пользователь может заказать у владельца сети необходимый уровень качества обслуживания, что включает: CIR – согласованная информационная скорость, с

которой сеть будет передавать данные; Bc- максимальное количество байтов, которое сеть будет передавать сеть от этого пользователя за интервал времени T; Be- максимальное количество байтов, которое сеть будет передавать сверх установленного значения Bc за интервал времени T.

Важной особенностью сетей frame relay является так же то, что производители оборудования стремятся поддержать передачу голоса. Магистральные коммутаторы сети frame relay передают голосовые кадры в первую очередь.

Отметим, что использование виртуальных каналов для построения сети имеет недостаток при большом количестве точек доступа к сети необходимо строить большое количество виртуальных каналов, который необходимо оплачивать отдельно. В сетях TCP/IP оплачивается количество точек доступа, а не количество связей между ними.

Тем не менее сети frame relay можно успешно использовать для объединения локальных сетей.

Технология ATM

Технология ATM (ATM- Asynchronous Transfer Mode) является самой современной иперспективной технологией глобальных сетей и разрабатывается как единый транспорт для с интеграцией всех возможных услуг. По замыслу разработчиков технология ATM сможет обеспечить следующие возможности:

- передачу в рамках одной сети компьютерного и мультимедийного (голос, видео) трафика, причем каждому виду трафика качество обслуживания будет соответствовать его потребностям;
- иерархию скоростей передачи данных от десятков Мбит/с до нескольких Гбит/с с гарантированной пропускной способностью для приложений;
- общие транспортные протоколы для локальных и глобальных сетей;
- сохранение существующих физических каналов связи и протоколов;
- взаимодействие с протоколами локальных и глобальных сетей: IP, ISDN, Ethernet, Token Ring и др.

Технология ATM совмещает в себе две технологии- коммутации пакетов и коммутации каналов. От первой технологии она заимствует передачу данных в виде адресуемых пакетов, а от второй использование пакетов небольшого фиксированного размера в результате чего задержки в сети становятся более предсказуемыми. Основные стандарты технологии ATM были приняты в 1993 году и работы по их разработке активно продолжаются.

Трафик компьютерных сетей имеет ярко выраженный асинхронный характер и пульсирующий характер, т.к. каждый компьютер посылает свою информацию в непредсказуемые заранее (случайные) моменты времени. Трафик компьютерной сети очень чувствителен к потерям данных, так как их необходимо восстанавливать за счет повторной передачи.

Мультимедийный трафик, передающий голос или изображение, наоборот характеризуется низкой пульсацией и высокой чувствительностью к задержкам передачи данных. Кроме этого указанные трафики имеют различные размеры предаваемых пакетов. Пакеты, содержащие компьютерные данные, могут иметь пакет длиной до 4500 байт, при передаче которого через коммутатор может произойти значительная задержка пакетов с голосовыми данными. Поэтому в технологии ATM любой вид трафика передается пакетами фиксированной и очень маленькой длины в 53 байта. Пакеты ATM называются ячейками (cell). Поле данных ячейки занимает 48 байт, а заголовок 5 байт.

Кроме стандартизации и выбора одного и того же размера ячейки для любого вида трафика в технологии ATM реализуется важнейшее требование, предъявляемое к компьютерным сетям- *заказ пропускной способности и качества обслуживания*. (частично реализованного в сетях frame relay).

Магистраль ATM обеспечивает большие скорости передачи данных и может обеспечить связь между отдельными городами или даже странами. Массовое применению технологии ATM в локальных сетях сдерживается ее высокой стоимостью по сравнению с инвестициями в существующие технологии локальных сетей.

Тема 20. Сеть Интернет

Основные определения

24 октября 1995 года Федеральный сетевой совет (FNC), США, единодушно одобрил резолюцию, определяющую термин "Интернет" Это определение разрабатывалось при участии специалистов в области сетей и в области прав на интеллектуальную собственность.

Интернет — это глобальная информационная система, которая:

1. логически взаимосвязана пространством глобальных уникальных адресов, основанных на Интернет-протоколе (IP) или на последующих расширениях или преемниках IP;
2. способна поддерживать коммуникации с использованием семейства Протокола управления передачей, который называется Интернет-протоколом (TCP/IP) или его последующих расширений/преемников и/или других IP-совместимых протоколов;
3. обеспечивает, использует или делает доступной, на общественной или частной основе, высокоуровневые сервисы, надстроенные над описанной здесь коммуникационной и иной связанной с ней инфраструктурой.

Как видно из определения, в основе сети Интернет лежит использование протокола сетевого уровня, IP- протокола, над которым должны работать протоколы более высокого уровня, в первую очередь TCP – протокол.

Следует отметить, что революционизирующее влияние Интернет на мир компьютеров и коммуникаций не имеет исторических аналогов. Изобретение телеграфа, телефона, радио и компьютера подготовило почву для происходящей ныне их беспрецедентной интеграции. Интернет одновременно является и средством общемирового вещания, и механизмом распространения информации, и средой для сотрудничества и общения людей и компьютеров, охватывающей весь земной шар.

Интернет представляет собой один из наиболее успешных примеров того, какую пользу могут принести долгосрочные вложения и поддержка исследований и разработки информационной инфраструктуры. Начиная с ранних исследований в области пакетной коммутации, правительства различных стран, промышленность и академическая наука оставались партнерами в развитии и развертывании этой новой сетевой технологии.

В историческом развитии сети Интернет можно выделить четыре различных аспекта:

- технологическая эволюция исследований по пакетной коммутации;
- развитие методов и средств эксплуатации и управления глобальной и сложной сетевой инфраструктурой;
- социальный аспект, приведший к образованию широкого сообщества пользователей;
- коммерциализация, характеризуемая чрезвычайно эффективным превращением результатов исследований в развернутую, широко доступную информационную систему

Зарождение Интернет

У истоков создания сети Интернет стояла группа ученых и инженеров Управления перспективных исследований и разработок Министерства обороны США – DARPA (Defence Advanced Research Agency), созданная в 1962 году под руководством Дж. Ликлайдера. Этим ученым впервые была сформулирована концепция «галактической сети», объединяющая огромное количество компьютеров, и с помощью которой каждый пользователь сможет быстро получить доступ к данным и программам, расположенным на любом компьютере. Эта концепция очень близка по духу современному состоянию Интернет. Одновременно появились работы Леонарда Клейнрока по теории коммутации пакетов (пакетной коммутации), в которых теоретически обосновывалось возможность создания компьютерных сетей на основе пакетной коммутации. В дальнейшем проведенные эксперименты показали, что компьютеры с разделением

времени могут успешно работать вместе, выполняя программы и осуществляя выборку на удаленной машине. Стало ясно и то, что телефонная система того времени с коммутацией соединений абсолютно непригодна для создания компьютерной сети.

В 1967 году появился проект первой компьютерной сети ARPANET, а в 1968 были доработана структура и спецификации этой сети, которая должна была работать по технологии коммутации пакетов. После разработки первого коммутатора пакетов компанией BBN, который назывался тогда интерфейсным процессором, появилась возможность провести соединения с их помощью нескольких компьютеров, находящихся на большом расстоянии друг от друга. В сентябре 1969 один из коммутаторов был установлен в Калифорнийском университете, к нему был подключен компьютер, а второй коммутатор с подключенным компьютером разместили в Стэнфордском исследовательском институте. Через месяц было послано первое компьютерное сообщение из Колифорнийского университета, которое было успешно принято в Стэнфорде. Двумя следующими узлами ARPANET стали университет города Санта- Барбара и Университет штат Юта.

Таким образом, к концу 1969 года первые четыре компьютера были объединены в первоначальную конфигурацию ARPANET В последующие годы число компьютеров, подключенных к ARPANET, быстро росло.

Одновременно велись работы по созданию функционально полного протокола межкомпьютерного взаимодействия и другого сетевого программного обеспечения. В декабре 1970 года Сетевая рабочая группа (Network Working Group, NWG) завершила работу над первой версией протокола, получившего название Протокол управления сетью (Network Control Prtocol, NCP). После того, как в 1971- 1972 годах этот протокол был реализован на всех узлах ARPANET, пользователи сети смогли приступить к разработке приложений работающих над этим протоколом.

В марте 1972 года появилось *первое такое приложение – электронная почта*. Создателем программы электронной почты стал сотрудник вышеупомянутой компании BBN Рэй Томлисон (Ray Tomlinson), он же предложил использовать значок @ («собака»). Для своего времени электронная почта стала тем, же чем в наши дни является служба WWW- исключительно мощным катализатором роста всех видов межперсональных потоков данных.

Концепция объединения сетей

Интернет основывается на идее существования множества независимых сетей почти произвольной архитектуры, начиная от ARPANET. Интернет в современном понимании воплощает ключевой технический принцип *открытости сетевой архитектуры*. При подобном подходе архитектура и техническая реализация отдельных сетей не навязываются извне - они могут свободно выбираться

поставщиком сетевых услуг при сохранении возможности объединения с другими сетями посредством сетевого уровня.

Открытая сетевая архитектура подразумевает, что отдельные сети могут проектироваться и разрабатываться независимо, со своими уникальными интерфейсами, предоставляемыми пользователям и/или другим поставщикам сетевых услуг, включая услуги Интернет. При проектировании каждой сети могут быть приняты во внимание специфика окружения и особые требования пользователей. Вообще говоря, не накладывается никаких ограничений на типы объединяемых сетей или их территориальный масштаб.

Как уже указывалось выше, в сети ARPANET использовался протокол NCP.

Однако NCP не содержал средств для адресации сетей и отдельных машин. В обеспечении сквозной надежности протокол NCP полагался на хорошие линии связи.. Если какие-то пакеты терялись, протокол и поддерживаемые им приложения должны были остановиться. В модели NCP отсутствовало сквозное управление ошибками, поскольку ARPANET должна была являться единственной существующей сетью, причем настолько надежной, что от компьютеров не требовалось умение реагировать на ошибки. Таким образом, протокол NCP не соответствовал требованиям открытой сетевой архитектуры и требовал серьезной доработки.

Сотрудник DARPA Роберт Канн в 1972 году предложил разработать новую версию протокола, удовлетворяющую требованиям окружения с открытой сетевой архитектурой.

Этот протокол позднее будет назван ***Transmission Control Protocol/ Internet Protocol (TCP/IP — Протокол управления передачей/Межсетевой протокол)***.

В то время как NCP действовал в как драйвер устройства, новинка должна была в большей мере напоминать коммуникационный протокол.

В основе разработки нового протокола лежали четыре принципа:

- каждая сеть должна сохранять свою индивидуальность. При подключении к Интернет сети не должны подвергаться внутренним переделкам;

- передача пакетов должна идти по принципу "максимум возможного". Если пакет не прибыл в пункт назначения, источник должен вскоре повторно передать его;

- для связывания сетей должны использоваться черные ящики; позднее их назовут *шлюзами и маршрутизаторами*.

На локальном уровне не должно существовать глобальной системы управления.

Самыми первыми результатами по реализации указанных принципов стало:

- Общение между двумя компьютерами логически должно представляться как обмен непрерывными последовательностями байт. Для идентификации байта используется его позиция в последовательности.

- Управление потоком данных осуществляется на основе механизмов подтверждений. Получатель может выбирать, когда посылать подтверждение, распространяющееся на все полученные к этому моменту пакеты.

В публикациях того времени по объединению сетей (начало 70 –х годов) первоначально описывался один протокол, названный TCP. Он предоставлял все услуги по транспортировке и перенаправлению данных в Интернет. Планировалось, что протокол TCP будет поддерживать целый диапазон транспортных сервисов. Затем, однако, протокол TCP был раздел на два протокола — простой IP, обслуживающий только адресацию и перенаправление отдельных пакетов, и отдельный TCP, имеющий дело с такими аспектами, как управление потоком данных и нейтрализация потери пакетов. Для приложений, не нуждавшихся в услугах TCP, была добавлена альтернатива — Пользовательский дэйтаграммный протокол (User Datagram Protocol, UDP), открывающий прямой доступ к базовым сервисам уровня IP с приложений верхнего уровня.

Ключевая концепция создания Интернет состояла в том, что объединение сетей проектировалось не для какого-то одного приложения, но как универсальная инфраструктура, над которой могут быть надстроены новые приложения. Основой этих приложений являлся протокол TCP / IP.

Широкое распространение в 1980-е годы локальных сетей, персональных компьютеров и рабочих станций дало толчок бурному росту Интернет. Технология Ethernet, разработанная в 1973 году фирмой Xerox PARC, в наши дни является доминирующей сетевой технологией в Интернет, а персональные компьютеры и рабочие станции стали доминирующими компьютерами.

Рост Интернет вызвал важные изменения и в вопросах управления. Чтобы сделать сеть более дружественной для человека, компьютерам были присвоены имена, делающие ненужным запоминание числовых адресов. Пол Мокапетрис (Paul Mockapetris) из Института информатики Университета Южной Калифорнии придумал доменную систему имен (Domain Name System, DNS). DNS позволила создать масштабируемый распределенный механизм для отображения иерархических имен компьютеров (например, www.acm.org) в Интернет-адреса.

Еще одной особенностью, вызванной ростом Интернет, стало внесение изменений в программное обеспечение. Протокол TCP/IP стал встраиваться в существующие операционные системы Unix.

В целом стратегия встраивания протоколов Интернет TCP/IP в самую распространенную операционную систему, явилась одним из ключевых элементов успешного и повсеместного распространения Интернет.

Протокол TCP/IP был принят в качестве военного стандарта в 1980 году. Это позволило военным начать использование технологической базы Интернет и, в

конце концов, привело к разделению на военное и гражданское Интернет-сообщества. К 1983 году ARPANET использовало значительное число военных исследовательских, разрабатывающих и эксплуатирующих организаций.

Кроме этого к 1985 году технологии Интернет поддерживались широкими кругами исследователей и разработчиков. Интернет начинали использовать для повседневных компьютерных коммуникаций люди самых разных категорий. Особую популярность завоевала электронная почта, работавшая на разных платформах. Совместимость различных почтовых систем продемонстрировала выгоды массовых электронных коммуникаций между людьми.

Громадным шагом в развитии Интернет стала разработка в 1989 году Тимом Бернерсом-Ли гипертекстовой среды, а также разработка им первого Web-браузера, который назывался World Wide Web.

17 мая 1991 года на вычислительных системах Европейской физической лаборатории CERN (European Organization for Nuclear research) была установлена окончательная версия первого в мире Web-сервера.

Создание инфраструктуры Интернет

К середине 1970-х годов компьютерные сети начали расти, как грибы после дождя. Министерство энергетики США сначала создало сеть MFENet в интересах исследователей термоядерного синтеза с магнитным удержанием, затем специалисты в области физики высоких энергий получили сеть HEPNet, для астрофизиков из NASA построили сеть SPAN, национальный научный фонд (NSF), США, развернул сеть CSNET, объединившую специалистов по информатике из академических и промышленных кругов. Указанные сети должны были использоваться замкнутым сообществом специалистов; как правило, этим работа сетей и ограничивалась. Особой потребности в совместимости сетей не было; соответственно, не было и самой совместимости. Важным шагом по объединению сетей стало в 1985 году важное решение об обязательном использовании в NSFNet протокола TCP/IP.

Размах сети NSFNet, воспринимаемой уже как сеть Интернет Размеры ее финансирования составили 200 миллионов долларов за период с 1986 по 1995 год. В сочетании с качеством TCP/IP протоколов это привело к тому, что в начале 90-х семейство TCP/IP вытеснило или значительно потеснило во всем мире большинство других протоколов глобальных компьютерных сетей. К 1990 году окончательно разукомплектовали сеть ARPANET, которая не могла уже конкурировать с новыми технологиями Интернет. Протокол IP уверенно становился доминирующим сервисом транспортировки данных в глобальной информационной инфраструктуре.

Использование масок в IP- адресации

Основной недостаток использования классов IP- адресов напрямую состоит в том, что если организация имеет несколько сетевых номеров, то все компьютеры вне сети имеют доступ к этим адресам и сеть организации становится прозрачной.

Для устранения указанного недостатка адресное пространство сети разбивается на более мелкие непересекающиеся пространства – подсети (subnet). С каждой из подсетей можно работать как с обычной TCP/IP – сетью.

Разбивка адресного пространства на подсети осуществляется с помощью *масок***.**

Маска- это число, которое используется в паре с IP- адресом; двоичная запись маски содержит единицы в тех разрядах, которые должны в IP- адресах интерпретироваться как номер сети. Единицы в маске должны представлять непрерывную последовательность.

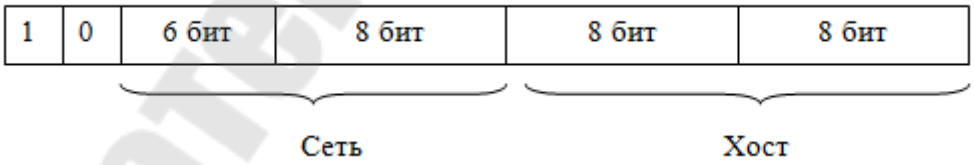
Для стандартных классов маски имеют следующие значения:

- Класс А – 11111111.00000000.00000000.00000000 (255.0.0.0)
- Класс В - 11111111.11111111.00000000.00000000 (255.255.0.0)
- Класс С - 11111111.11111111.11111111.00000000 (255.255.255.0)

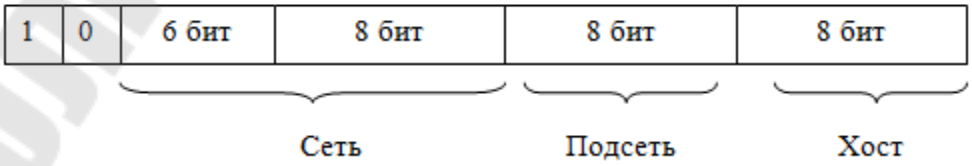
Рассмотрим, каким образом маска преобразует IP- адреса.

Пусть организация получила один IP- адрес класса В. Как известно, для сетей класса В первые два байта являются номером сети, а два остальные байта определяют номер узла. Для организации подсетей и их нумерации используются разряды байтов номеров узлов. В самом простом случае для нумерации подсетей используется первый байт номера узла.

Адрес до преобразования выглядел следующим образом:



После организации подсети IP- адрес стал выглядеть:

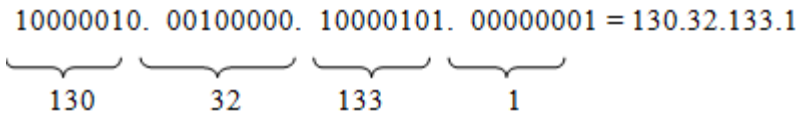


Задавая в третьем байте номера подсети, можно разбивать сеть на отдельные подсети и присваивать номера узлов внутри подсети. В этом случае нумерация

узлов внутри подсетей является локальным для организации и не видна во внешней сети. Все компьютеры вне организации видят одну большую IP- сеть и они должны поддерживать только маршруты доступа к шлюзам, соединяющим сеть организации с внешним миром.

Пример

IP- адрес сети класса В задан в виде:



а) Маска не используется. В этом случае номером сети являются первые два байта и определяют сеть 130.32.0.0, а номер узла равен 0.0.132.1

б) Используется маска:

11111111.11111111.10000000.00000000 = 255.255.128.0

В этом случае наложение маски на IP- адрес дает новое число, интерпретируемое как номер сети:

10000010. 00100000. 10000000. 00000000 = 130.32.128.0

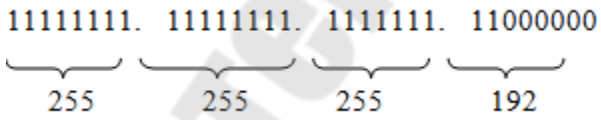
Номер узла в этой сети становится 0.0.5.1

Как видно из примера, снабжая IP-адреса маской, можно отказаться от понятий классов адресов и сделать более гибкой систему адресации сетей.

Пример

Пусть в сети работают два компьютера, имеющие два соответствующие IP-адреса: 210.20.30.193 и 210.20.30.70. Для разделения указанных компьютеров в две разные подсети используем маску 255.255.255.192

В двоичной форме маска имеет вид:



Двоичный адрес первого компьютера:

11010010. 00010100. 00011110. 11000001

210 20 30 193

Двоичный адрес второго компьютера:

11010010. 00010100. 00011110. 01000110

210 20 30 70

Накладывая маску на адрес первого компьютера, получим его новый адрес:

11010010. 00010100. 00011110. 11 000001

210 20 30 / 6

Подсеть № 3

Накладывая маску на адрес второго компьютера, получим его новый адрес:

11010010. 00010100. 00011110. 01 000110

210 20 30 / 6

Подсеть № 1

Таким образом, сеть с помощью маски разбилась на две подсети, номер второго компьютера в подсети стал равным шести.

Следует отметить, что в настоящее время наблюдается дефицит IP- адресов, выделяемых организацией InterNIC. Очень трудно получить адрес класса B и практически невозможно стать обладателем адреса класса A. Если же IP- сеть создана для работы в автономном режиме, без связи с Интернет, то администратор сети сам произвольно назначает номер. Но даже в этой ситуации в стандартах Интернет определены несколько диапазонов адресов, не рекомендуемых для использования в локальных сетях. Эти адреса не обрабатываются маршрутизаторами Интернет ни при каких условиях. Для сетей класса A – это сеть 10.0.0.0, в классе B- это диапазон из 16 номеров сетей 172.16.0.0 – 172.31.0.0, в классе C – это диапазон из 255 сетей – 192.168.0.0 – 192.168.255.0.

Для разрешения проблемы дефицита адресов осуществляется переход на новую версию IP- протокола- протокол IPv6, в котором резко расширяется адресное пространство за счет 16- байтных адресов.

Протокол IPv6, как развитие транспортных средств IP- протокола

Указанный протокол решает принципиальную проблему нехватки IP-адресов посредством использования 128- разрядных адресов вместо 32 – разрядных адресов,

благодаря чему адресное пространство расширяется в 296 раз. Результатом этого будет то, что любой житель Земли может получить в сове распоряжение несколько IP- адресов, новое количество адресов позволит подключить к сети свыше 1 квадрильона компьютеров в 1 триллионе сетей.

Адреса в IPv6 – протоколе разделяются на три типа: *обычные, групповые и нечеткие.*

Пакет с обычным адресом передается конкретному адресату, в то время как пакет с групповым адресом доставляется всем членам группы. Пакет с нечетким адресом доставляется только ближайшему члену данной группы.

В IPv6 128 разрядные адреса записываются в виде восьми 16- разрядных целых чисел, разделенных двоеточие. Каждое число представлено шестнадцатеричными цифрами, разделенными двоеточиями. Другими словами, необходимо вводить 32 шестнадцатеричные цифры для задания IP- адреса. IPv6 – адрес может выглядеть так: 501A:0000:0000:0000:00FC:ABCD:3F1F:3D5A.

Переход от традиционных IP- адресов к IPv6 – адресам займет ни один год и старая адресация будет постепенно замещаться новыми программными продуктами и оборудованием, использующим IPv6- протокол.

Среди других новых свойств IPv6 – протокола можно отметить также более рациональную структуру формата заголовка пакета, увеличение производительности маршрутизаторов, работающих с этим протоколом, возможность маркировки потока данных, если их необходимо обрабатывать особым образом, аутентификацию дейтаграмм и д.р.

Система доменов DNS

Выше было установлено, что для обращения к хостам используются 32-разрядные IP- адреса. Поскольку при работе в сети Интернет использовать цифровую адресацию сетей крайне неудобно, то вместо цифр используются символьные имена, называемыми *доменными именами.* Доменом называется группа компьютеров, объединенных одним именем. Символьные имена дают пользователю возможность лучше ориентироваться в Интернет, поскольку запомнить имя всегда проще, чем цифровой адрес.

На заре создания Интернет соответствия между именами хостов и их IP-адресами были размещены в единственном файле, который назывался Hosts.txt, который размещался на компьютере в центре InterNIC. Этот файл передавался по всем хостам еще совсем тогда крохотной сети. Стремительный рост Интернет заставил выработать новую концепцию механизма разрешения имен. С этой целью была разработана специальная система DNS (Domain Name System), для реализации которой был создан специальный сетевой протокол DNS. Начальные попытки создать единую копию целой базы данных имен и адресов оказались тщетными из-за громадного объема информации. Было принято решение строить распределенную

базу данных, а для увеличения производительности использовать механизм локального кэширования (сохранения в локальной базе данных). Доступ к распределенной базе данных не зависит ни от аппаратной платформы хоста, ни от коммутационной системы. Доступ к базе данных должны иметь все пользователи Интернет. Администрирование базы данных DNS возлагается на каждую организацию, которая подключается к Интернет. Организация должна инсталлировать свой собственный компьютер -сервер разрешения имен и ту часть распределенной базы данных, содержащей информацию о домене хостов данной организации. Сервер должен обслуживать хосты внутри организации и предоставлять доступ к базе данных этой организации извне.

Структура баз данных в системе DNS имеет иерархический вид, аналогичный иерархии файлов, принятой во многих файловых системах. Дерево имен начинается с корня, затем следует старшая символьная часть имени, вторая часть имени и т.д. Младшая часть имени соответствует конечному узлу сети. Все имена разделяются точками, причем иерархия задается справа налево, например, www.bseu.minsk.by

По имени можно получить информацию о профиле организации или ее местоположении. Шесть доменов высшего уровня определены следующим образом:

- gov – правительственные организации;
- mil – военные организации;
- edu – образовательные организации;
- com - коммерческие организации;
- org- общественные организации;
- net – организации, предоставляющие сетевые услуги, как правило, региональные сетевые организации.

Кроме того, все страны мира имеют свое собственное символьное имя, обозначающий домен верхнего уровня этой страны. Например, de – Германия, us – США, ru- Россия, by – Беларусь и т.д. Таким образом, адрес www.cdo.bseu.minsk.by означает, что компьютер дистанционного образования cdo находится в группе компьютеров (в домене) Белорусского государственного экономического университета bseu, в домене minsk в Республике Беларусь. Графически DNS можно представить в виде дерева, как на рисунке 52.

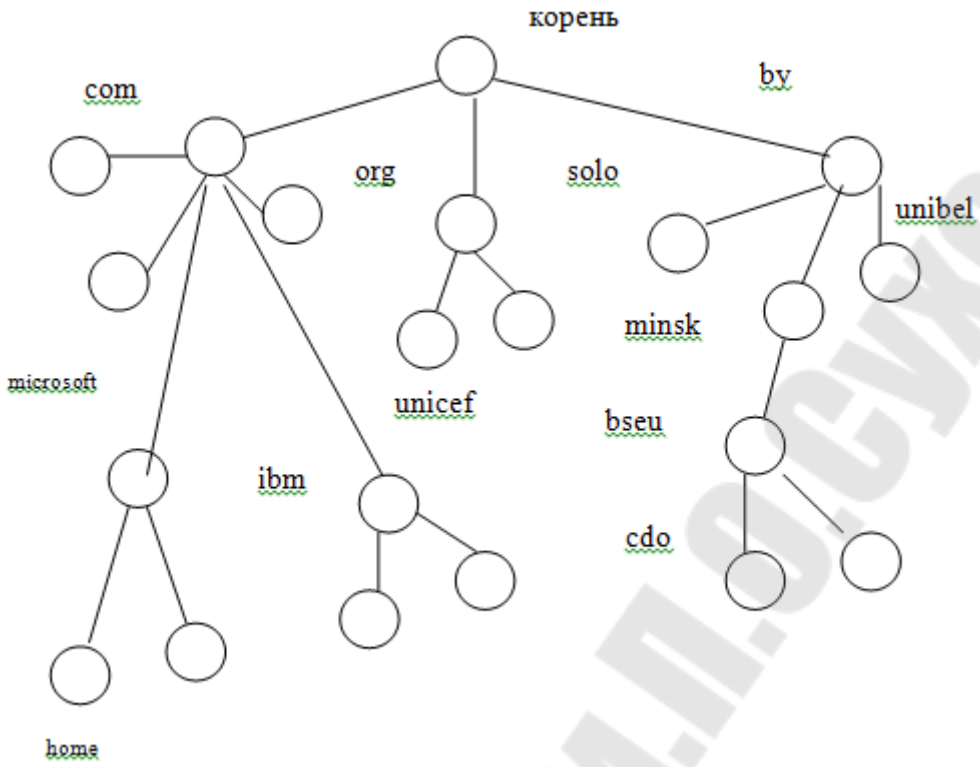


Рисунок 52 – Дерево системы DNS

DNS имеет три основные компоненты:

- Пространство имен домена (domain name space) и записи базы данных DNS (resource records).Они определяют структуру имен «дерева» и данных, связанных с этими именами. Запрос по данному имени возвратит IP- адрес хоста.
- Сервера имен (name servers). Сервера имен – это специальные компьютеры со специальными серверными программами, обрабатывающие информацию имен и данных имен. Сервер управляет всей информацией подчиненной ему области имен и данных домена. При обращении за информацией, который данный сервер не обслуживает, он должен или переправить запрос серверу, обслуживающему эту информацию, или стоящему на следующей ступени иерархии. Сервер, в распоряжении которого находится определенная часть информации об именах, является владельцем (authority) имен домена, а граница владения называется зоной (zone). Зоны строятся не на основе принадлежности какой-либо части данных к определенной организации, а распределяются автоматически серверами имен и должны обеспечить полную адресацию хостов.
- Программы разрешения имен (resolves). Эти программы возвращают информацию, хранящуюся в базе данных имен домена по запросу пользователя. Пользователь взаимодействует с пространством имен через указанные программы. Как правило эти программы реализуются в виде системного модуля, напрямую связанного с пользовательской программой, поэтому не требуется ни какого дополнительного протокола обмена.

Основным предназначением системы имен доменов является обеспечение механизма именования ресурсов. Этот механизм должен эффективно работать с различными хостами, сетями, семействами протоколов и типами организаций. Описанная выше структура DNS позволяет решать проблему адресации отдельных модулей изолировано, и, тем самым, создает универсальную модульную архитектуру.

Пользователь взаимодействует с пространством имен через программы разрешения. Для работы программ разрешения необходимо обращаться к серверам имен на других хостах, что может давать задержки от миллисекунд до нескольких секунд. Поэтому одной из важнейших свойств программ разрешения имен является возможность устранения сетевых задержек ответов. При этом используется механизм *кэширования результатов запросов имен.* Этот механизм ускоряет процесс определения имен, так в КЭШ-памяти накапливается информация о всех предыдущих именах , к которым обращалась программа.

Наиболее упрощенный и распространенный принцип работы такой программы с серверами имен показан на рисунке 53.

Рисунок 53 – Принцип работы с серверами имен

Программа пользователя запрашивает имя хоста и передает этот запрос программе разрешения имен. В первую очередь программа разрешения имен обращается за необходимым IP- адресом в собственную КЭШ- память. Если требуемого имени а КЭШ – памяти не находится, программа разрешения имен обращается к удаленному серверу имен, В случае нахождения необходимого имени, программа возвращает пользователю требуемый IP- адрес, одновременно записывая его в КЭШ- память.

Система DNS требует, чтобы доступ к информации определенной зоны мог быть осуществлен с нескольких серверов доменов. Существует механизм предоставления пользователям различных доменов совместного использования информации путем установления *доверительных отношений* между доменами. При

этом доверительные отношения могут быть как *двухсторонними,* так и *односторонними*.

При двухсторонних доверительных отношениях пользователь любого из двух доменов имеет доступ к информации, находящихся на соседнем домене.

При односторонних доверительных отношениях пользователь, находящийся в доверяемом домене, имеет доступ к серверам доменам доверителя, но не наоборот.

РАЗДЕЛ 7. БЕЗОПАСНОСТЬ КОМПЬЮТЕРНЫХ СЕТЕЙ

Тема 21. Защита информации в локальных и глобальных сетях.

Одной из наиболее очевидных причин нарушения системы защиты является умышленный несанкционированный доступ (НСД) к конфиденциальной информации со стороны нелегальных пользователей и последующие нежелательные манипуляции с этой информацией. Защита информации – это комплекс мероприятий, проводимых с целью предотвращения утечки, хищения, утраты, несанкционированного уничтожения, искажения, модификации (подделки), несанкционированного копирования, блокирования информации и т.п. Поскольку утрата информации может происходить по сугубо техническим, объективным и неумышленным причинам, под это определение попадают также и мероприятия, связанные с повышением надежности сервера из-за отказов или сбоев в работе винчестеров, недостатков в используемом программном обеспечении и т.д.

Следует заметить, что наряду с термином "защита информации" (применительно к компьютерным сетям) широко используется, как правило, в близком значении, термин "компьютерная безопасность".

Переход от работы на персональных компьютерах к работе в сети усложняет защиту информации по следующим причинам:

большое число пользователей в сети и их переменный состав. Защита на уровне имени и пароля пользователя недостаточна для предотвращения входа в сеть посторонних лиц;

значительная протяженность сети и наличие многих потенциальных каналов проникновения в сеть;

уже отмеченные недостатки в аппаратном и программном обеспечении, которые зачастую обнаруживаются не на предпродажном этапе, называемом бета-тестированием, а в процессе эксплуатации. В том числе неидеальны встроенные средства защиты информации даже в таких известных и "мощных" сетевых ОС, как Windows NT или NetWare.

Остроту проблемы, связанной с большой протяженностью сети для одного из ее сегментов на коаксиальном кабеле, иллюстрирует рис. 9.1. В сети имеется много физических мест и каналов несанкционированного доступа к информации в сети. Каждое устройство в сети является потенциальным источником электромагнитного излучения из-за того, что соответствующие поля, особенно на высоких частотах, экранированы неидеально. Система заземления вместе с кабельной системой и сетью электропитания может служить каналом доступа к информации в сети, в том числе на участках, находящихся вне зоны контролируемого доступа и потому особенно уязвимых. Кроме электромагнитного излучения, потенциальную угрозу представляет бесконтактное электромагнитное воздействие на кабельную систему. Безусловно, в случае использования проводных соединений типа коаксиальных кабелей или витых пар, называемых часто медными кабелями, возможно и

непосредственное физическое подключение к кабельной системе. Если пароли для входа в сеть стали известны или подобраны, становится возможным несанкционированный вход в сеть с файл-сервера или с одной из рабочих станций. Наконец возможна утечка информации по каналам, находящимся вне сети:

– хранилище носителей информации,

– элементы строительных конструкций и окна помещений, которые образуют каналы утечки конфиденциальной информации за счет так называемого микрофонного эффекта,

– телефонные, радио-, а также иные проводные и беспроводные каналы (в том числе каналы мобильной связи).

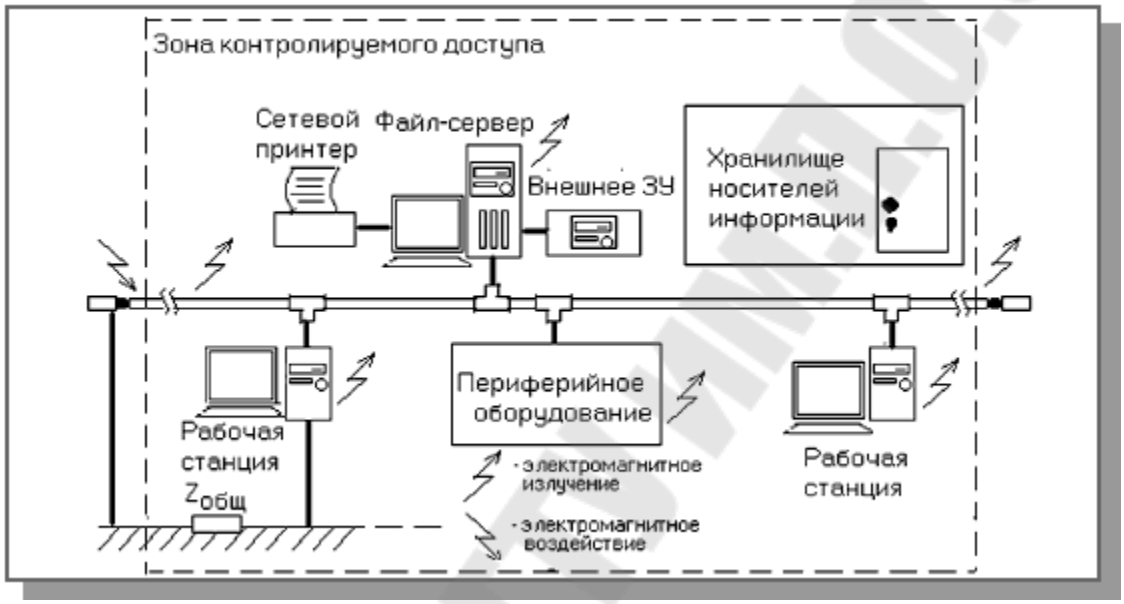


Рисунок 54 – Места и каналы возможного несанкционированного доступа к информации в компьютерной сети

Любые дополнительные соединения с другими сегментами или подключение к Интернет порождают новые проблемы. Атаки на локальную сеть через подключение к Интернету для того, чтобы получить доступ к конфиденциальной информации, в последнее время получили широкое распространение, что связано с недостатками встроенной системы защиты информации в протоколах TCP/IP. Сетевые атаки через Интернет могут быть классифицированы следующим образом:

1. Сниффер пакетов (sniffer – в данном случае в смысле фильтрация) – прикладная программа, которая использует сетевую карту, работающую в режиме promiscuous (не делающий различия) mode (в этом режиме все пакеты, полученные по физическим каналам, сетевой адаптер отправляет приложению для обработки).

2. IP-спуфинг (spoof – обман, мистификация) – происходит, когда хакер, находящийся внутри корпорации или вне ее, выдает себя за санкционированного пользователя.

3. Отказ в обслуживании (Denial of Service – DoS). Атака DoS делает сеть недоступной для обычного использования за счет превышения допустимых пределов функционирования сети, операционной системы или приложения.

4. Парольные атаки – попытка подбора пароля легального пользователя для входа в сеть.

5. Атаки типа Man-in-the-Middle – непосредственный доступ к пакетам, передаваемым по сети.

6. Атаки на уровне приложений.

7. Сетевая разведка – сбор информации о сети с помощью общедоступных данных и приложений.

8. Злоупотребление доверием внутри сети.

9. Несанкционированный доступ (НСД), который не может считаться отдельным типом атаки, так как большинство сетевых атак проводятся ради получения несанкционированного доступа.

10. Вирусы и приложения типа "троянский конь".

Классификация средств защиты информации

Защита информации в сети на рис. 53. может быть улучшена за счет использования специальных генераторов шума, маскирующих побочные электромагнитные излучения и наводки, помехоподавляющих сетевых фильтров, устройств зашумления сети питания, скремблеров (шифраторов телефонных переговоров), подавителей работы сотовых телефонов и т.д. Кардинальным решением является переход к соединениям на основе оптоволокна, свободным от влияния электромагнитных полей и позволяющим обнаружить факт несанкционированного подключения.

В целом средства обеспечения защиты информации в части предотвращения преднамеренных действий в зависимости от способа реализации можно разделить на группы:

Технические (аппаратные) средства. Это различные по типу устройства (механические, электромеханические, электронные и др.), которые аппаратными средствами решают задачи защиты информации. Они либо препятствуют физическому проникновению, либо, если проникновение все же состоялось, доступу к информации, в том числе с помощью ее маскировки. Первую часть задачи решают замки, решетки на окнах, защитная сигнализация и др. Вторую – упоминавшиеся выше генераторы шума, сетевые фильтры, сканирующие радиоприемники и множество других устройств, "перекрывающих" потенциальные каналы утечки информации или позволяющих их обнаружить. Преимущества технических средств связаны с их надежностью, независимостью от субъективных факторов, высокой устойчивостью к модификации. Слабые стороны – недостаточная гибкость, относительно большие объем и масса, высокая стоимость.

Программные средства включают программы для идентификации пользователей, контроля доступа, шифрования информации, удаления остаточной

(рабочей) информации типа временных файлов, тестового контроля системы защиты и др. Преимущества программных средств – универсальность, гибкость, надежность, простота установки, способность к модификации и развитию. Недостатки – ограниченная функциональность сети, использование части ресурсов файл-сервера и рабочих станций, высокая чувствительность к случайным или преднамеренным изменениям, возможная зависимость от типов компьютеров (их аппаратных средств).

Смешанные аппаратно-программные средства реализуют те же функции, что аппаратные и программные средства в отдельности, и имеют промежуточные свойства.

Организационные средства складываются из организационно-технических (подготовка помещений с компьютерами, прокладка кабельной системы с учетом требований ограничения доступа к ней и др.) и организационно-правовых (национальные законодательства и правила работы, устанавливаемые руководством конкретного предприятия). Преимущества организационных средств состоят в том, что они позволяют решать множество разнородных проблем, просты в реализации, быстро реагируют на нежелательные действия в сети, имеют неограниченные возможности модификации и развития. Недостатки – высокая зависимость от субъективных факторов, в том числе от общей организации работы в конкретном подразделении.

По степени распространения и доступности выделяются программные средства, поэтому далее они рассматриваются более подробно (см. "Стандартные методы шифрования и криптографические системы" и "Программные средства защиты информации"). Другие средства применяются в тех случаях, когда требуется обеспечить дополнительный уровень защиты информации.

Шифрование данных представляет собой разновидность программных средств защиты информации и имеет особое значение на практике как единственная надежная защита информации, передаваемой по протяженным последовательным линиям, от утечки. Шифрование образует последний, практически непреодолимый "рубеж" защиты от НСД. Понятие "шифрование" часто употребляется в связи с более общим понятием криптографии. Криптография включает способы и средства обеспечения конфиденциальности информации (в том числе с помощью шифрования) и аутентификации. Конфиденциальность – защищенность информации от ознакомления с ее содержанием со стороны лиц, не имеющих права доступа к ней. В свою очередь аутентификация представляет собой установление подлинности различных аспектов информационного взаимодействия: сеанса связи, сторон (идентификация), содержания (имитозащита) и источника (установление авторства с помощью цифровой подписи).

Число используемых программ шифрования ограничено, причем часть из них являются стандартами де-факто или де-юре. Однако даже если алгоритм шифрования не представляет собой секрета, произвести дешифрование

(расшифрование) без знания закрытого ключа чрезвычайно сложно. Это свойство в современных программах шифрования обеспечивается в процессе многоступенчатого преобразования исходной открытой информации (plain text в англоязычной литературе) с использованием ключа (или двух ключей – по одному для шифрования и дешифрования). В конечном счете, любой сложный метод (алгоритм) шифрования представляет собой комбинацию относительно простых методов.

Классические алгоритмы шифрования данных

Имеются следующие "классические" методы шифрования:

- подстановка (простая – одноалфавитная, многоалфавитная однопетлевая, многоалфавитная многопетлевая);
- перестановка (простая, усложненная);
- гаммирование (смешивание с короткой, длинной или неограниченной маской).

Устойчивость каждого из перечисленных методов к дешифрованию без знания ключа характеризуется количественно с помощью показателя Sк, представляющего собой минимальный объем зашифрованного текста, который может быть дешифрован посредством статистического анализа.

Подстановка предполагает использование альтернативного алфавита (или нескольких) вместо исходного.

Стандартные методы шифрования (национальные или международные) для повышения степени устойчивости к дешифрованию реализуют несколько этапов (шагов) шифрования, на каждом из которых используются различные "классические" методы шифрования в соответствии с выбранным ключом (или ключами). Существуют две принципиально различные группы стандартных методов шифрования:

- шифрование с применением одних и тех же ключей (шифров) при шифровании и дешифровании (симметричное шифрование или системы с закрытыми ключами – private-key systems);
- шифрование с использованием открытых ключей для шифрования и закрытых – для дешифрования (несимметричное шифрование или системы с открытыми ключами – public-key systems).

Строгое математическое описание алгоритмов стандартных методов шифрования слишком сложно. Для пользователей важны в первую очередь "потребительские" свойства различных методов (степень устойчивости к дешифрованию, скорость шифрования и дешифрования, порядок и удобство распространения ключей), которые и рассматриваются ниже.

Для дальнейшего повышения устойчивости к дешифрованию могут применяться последовательно несколько стандартных методов или один метод шифрования (но с разными ключами).

Стандартные методы шифрования и криптографические системы

Стандарт шифрования США DES (Data Encryption Standard – стандарт шифрования данных) относится к группе методов симметричного шифрования и действует с 1976 г. Число шагов – 16. Длина ключа – 64 бита, из которых 8 бит – проверочные разряды четности/нечетности. Долгое время степень устойчивости к дешифрованию этого метода считалась достаточной, однако в настоящее время он устарел. Вместо DES предлагается "тройной DES" – 3DES, в котором алгоритм DES используется 3 раза, обычно в последовательности "шифрование – дешифрование – шифрование" с тремя разными ключами на каждом этапе.

Надежным считается алгоритм IDEA (International Data Encryption Algorithm), разработанный в Швейцарии и имеющий длину ключа 128 бит.

Отечественный ГОСТ28147-89 – это аналог DES, но с длиной ключа 256 бит, так что его степень устойчивости к дешифрованию изначально существенно выше. Важно также и то, что в данном случае предусматривается целая система защиты, которая преодолевает "родовой" недостаток симметричных методов шифрования – возможность подмены сообщений. Такие усовершенствования, как имитовставки, хэш-функции и электронные цифровые подписи позволяют "авторизовать" передаваемые сообщения.

К достоинствам симметричных методов шифрования относится высокая скорость шифрования и дешифрования, к недостаткам – малая степень защиты в случае, если ключ стал доступен третьему лицу.

Довольно популярны, особенно при использовании электронной почты в Интернет, несимметричные методы шифрования или системы с открытыми ключами – public-key systems. К этой группе методов относится, в частности, PGP (Pretty Good Privacy – достаточно хорошая секретность). Каждый пользователь имеет пару ключей. Открытые ключи предназначены для шифрования и свободно рассылаются по сети, но не позволяют произвести дешифрование. Для этого нужны секретные (закрытые) ключи. Принцип шифрования в данном случае основывается на использовании так называемых односторонних функций. Прямая функция x >> f(x) легко вычисляется на основании открытого алгоритма (ключа). Обратное преобразование. f(x) >> x без знания закрытого ключа затруднено и должно занимать довольно длительное время, которое и определяет степень "трудновычислимости" односторонней функции.

Идея системы с открытыми ключами может быть пояснена следующим образом (табл. 11). Для шифрования сообщений можно взять обычную телефонную книгу, в которой имена абонентов расположены в алфавитном порядке и предшествуют телефонным номерам. У пользователя имеется возможность выбора соответствия между символом в исходном тексте и именем абонента, то есть это многоалфавитная система. Ее степень устойчивости к дешифрованию выше. Легальный пользователь имеет "обратный" телефонный справочник, в котором в первом столбце располагаются телефонные номера по возрастанию, и легко

производит дешифрование. Если же такового нет, то пользователю предстоит утомительное и многократное просматривание доступного прямого справочника в поисках нужных телефонных номеров. Это и есть практическая реализация трудно-вычислимой функции. Сам по себе метод шифрования на основе телефонных справочников вряд ли перспективен хотя бы из-за того, что никто не мешает потенциальному взломщику составить "обратный" телефонный справочник. Однако в используемых на практике методах шифрования данной группы в смысле надежности защиты все обстоит благополучно.

Таблица 11 - Пример шифрования в системе с открытыми ключами

Исходное слово	Выбранное имя абонента	Зашифрованное сообщение (телефонные номера)
S	Scott	3541920
A	Adleman	4002132
U	Ullman	7384502
N	Nivat	5768115
A	Aho	7721443

Другая известная система с открытыми ключами – RSA.

Несимметричные методы шифрования имеют преимущества и недостатки, обратные тем, которыми обладают симметричные методы. В частности, в несимметричных методах с помощью посылки и анализа специальных служебных сообщений может быть реализована процедура аутентификации (проверки легальности источника информации) и целостности (отсутствия подмены) данных. При этом выполняются операции шифрования и дешифрования с участием открытых ключей и секретного ключа данного пользователя. Таким образом, несимметричные системы можно с достаточным основанием отнести к полноценным криптографическим системам. В отличие от симметричных методов шифрования, проблема рассылки ключей в несимметричных методах решается проще – пары ключей (открытый и закрытый) генерируются "на месте" с помощью специальных программ. Для рассылки открытых ключей используются такие технологии как LDAP (Lightweight Directory Access Protocol – протокол облегченного доступа к справочнику). Рассылаемые ключи могут быть предварительно зашифрованы с помощью одного из симметричных методов шифрования.

Традиционные и обязательные для современных криптографических систем способы обеспечения аутентификации и проверки целостности получаемых данных (хэш-функции и цифровые подписи), которые реализуются непосредственными участниками обмена, не являются единственно возможными. Распространен также способ, осуществляемый с участием сторонней организации, которой доверяют все участники обменов. Речь идет об использовании так называемых цифровых

сертификатов – посылаемых по сети сообщений с цифровой подписью, удостоверяющей подлинность открытых ключей.

Программные средства защиты информации

Встроенные средства защиты информации в сетевых ОС доступны, но не всегда, как уже отмечалось, могут полностью решить возникающие на практике проблемы. Например, сетевые ОС NetWare 3.x, 4.x позволяют осуществить надежную "эшелонированную" защиту данных от аппаратных сбоев и повреждений. Система SFT (System Fault Tolerance – система устойчивости к отказам) компании Novell включает три основные уровня:

SFT Level I предусматривает, в частности, создание дополнительных копий FAT и Directory Entries Tables, немедленную верификацию каждого вновь записанного на файловый сервер блока данных, а также резервирование на каждом жестком диске около 2% от объема диска. При обнаружении сбоя данные перенаправляются в зарезервированную область диска, а сбойный блок помечается как "плохой" и в дальнейшем не используется.

SFT Level II содержит дополнительные возможности создания "зеркальных" дисков, а также дублирования дисковых контроллеров, источников питания и интерфейсных кабелей.

SFT Level III позволяет применять в локальной сети дублированные серверы, один из которых является "главным", а второй, содержащий копию всей информации, вступает в работу в случае выхода "главного" сервера из строя.

Система контроля и ограничения прав доступа в сетях NetWare (защита от несанкционированного доступа) также содержит несколько уровней:

- уровень начального доступа (включает имя и пароль пользователя, систему учетных ограничений – таких как явное разрешение или запрещение работы, допустимое время работы в сети, место на жестком диске, занимаемое личными файлами данного пользователя, и т.д.);

- уровень прав пользователей (ограничения на выполнение отдельных операций и/или на работу данного пользователя, как члена подразделения, в определенных частях файловой системы сети);

- уровень атрибутов каталогов и файлов (ограничения на выполнение отдельных операций, в том числе удаления, редактирования или создания, идущие со стороны файловой системы и касающиеся всех пользователей, пытающихся работать с данными каталогами или файлами);

- уровень консоли файл-сервера (блокирование клавиатуры файл-сервера на время отсутствия сетевого администратора до ввода им специального пароля).

Однако полагаться на эту часть системы защиты информации в ОС NetWare можно не всегда. Свидетельством тому являются многочисленные инструкции в Интернете и готовые доступные программы, позволяющие взломать те или иные элементы защиты от несанкционированного доступа.

То же замечание справедливо по отношению к более поздним версиям ОС NetWare (вплоть до последней 6-й версии) и к другим "мощным" сетевым ОС со встроенными средствами защиты информации (Windows NT, UNIX). Дело в том, что защита информации – это только часть тех многочисленных задач, которые решаются сетевыми ОС. Усовершенствование одной из функций в ущерб другим (при понятных разумных ограничениях на объем, занимаемый данной ОС на жестком диске) не может быть магистральным направлением развития таких программных продуктов общего назначения, которыми являются сетевые ОС. В то же время в связи с остротой проблемы защиты информации наблюдается тенденция интеграции (встраивания) отдельных, хорошо зарекомендовавших себя и ставших стандартными средств в сетевые ОС, или разработка собственных "фирменных" аналогов известным программам защиты информации. Так, в сетевой ОС NetWare 4.1 предусмотрена возможность кодирования данных по принципу "открытого ключа" (алгоритм RSA) с формированием электронной подписи для передаваемых по сети пакетов.

Специализированные программные средства защиты информации от несанкционированного доступа обладают в целом лучшими возможностями и характеристиками, чем встроенные средства сетевых ОС. Кроме программ шифрования и криптографических систем, существует много других доступных внешних средств защиты информации. Из наиболее часто упоминаемых решений следует отметить следующие две системы, позволяющие ограничить и контролировать информационные потоки.

Firewalls – брандмауэры (дословно firewall – огненная стена). Между локальной и глобальной сетями создаются специальные промежуточные серверы, которые инспектируют и фильтруют весь проходящий через них трафик сетевого/транспортного уровней. Это позволяет резко снизить угрозу несанкционированного доступа извне в корпоративные сети, но не устраняет эту опасность полностью. Более защищенная разновидность метода – это способ маскарада (masquerading), когда весь исходящий из локальной сети трафик посылается от имени firewall-сервера, делая локальную сеть практически невидимой.

Proxy-servers (proxy – доверенность, доверенное лицо). Весь трафик сетевого/транспортного уровней между локальной и глобальной сетями запрещается полностью – маршрутизация как таковая отсутствует, а обращения из локальной сети в глобальную происходят через специальные серверы-посредники. Очевидно, что при этом обращения из глобальной сети в локальную становятся невозможными в принципе. Этот метод не дает достаточной защиты против атак на более высоких уровнях – например, на уровне приложения (вирусы, код Java и JavaScript).

Тема 22. Безопасность ЛВС при взаимодействии с Интернет

В наше время в деятельности любого коммерческого предприятия очень большую важность имеет защита информации. Информация сегодня – ценные ресурс, от которого зависит как функционирование предприятия в целом, так и его конкурентоспособность. Угроз безопасности информационных ресурсов предприятия много – это и компьютерные вирусы, которые могут уничтожить важные данные, и промышленный шпионаж со стороны конкурентов преследующих своей целью получение незаконного доступа к информации представляющей коммерческую тайну, и много другое. Поэтому особое место приобретает деятельность по защите информации, по обеспечению информационной безопасности.

Информационная безопасность (англ. «Information security») – защищенность информации и соответствующей инфраструктуры от случайных или преднамеренных воздействий сопровождающихся нанесением ущерба владельцам или пользователям информации. Информационная безопасность – обеспечение конфиденциальности, целостности и доступности информации. Цель защиты информации – минимизация потерь, вызванных нарушением целостности или конфиденциальности данных, а также их недоступности для потребителей.

Основные типы угроз информационной безопасности: 1. Угрозы конфиденциальности – несанкционированный доступ к данным (например, получение посторонними лицами сведений о состоянии счетов клиентов банка). 2. Угрозы целостности – несанкционированная модификация, дополнение или уничтожение данных (например, внесение изменений в бухгалтерские проводки с целью хищения денежных средств). 3. Угрозы доступности – ограничение или блокирование доступа к данным (например, невозможность подключится к серверу с базой данных в результате DDoS-атаки).

Источники угроз: 1. Внутренние: а) ошибки пользователей и сисадминов; б) ошибки в работе ПО; в) сбои в работе компьютерного оборудования; г) нарушение сотрудниками компании регламентов по работе с информацией. 2. Внешние угрозы: а) несанкционированный доступ к информации со стороны заинтересованных организаций и отдельных лица (промышленный шпионаж конкурентов, сбор информации спецслужбами, атаки хакеров и т.п.); б) компьютерные вирусы и иные вредоносные программы; в) стихийные бедствия и техногенные катастрофы (например, ураган может нарушить работу телекоммуникационной сети, а пожар уничтожить сервера с важной информацией).

Методы обеспечения безопасности информации в ИС:

Препятствие - физическое преграждение пути злоумышленнику к защищаемой информации (например, коммерчески важная информация хранится на сервере внутри здания компании, доступ в которое имеют только ее сотрудники).

Управление доступом – регулирование использования информации и доступа к ней за счет системы идентификации пользователей, их опознавания, проверки полномочий и т.д. (например, когда доступ в отдел или на этаж с компьютерами, на которых хранится секретная информация, возможен только по специальной карточке-пропуску. Или когда каждому сотруднику выдается персональный логин и пароль для доступа к базе данных предприятия с разными уровнями привилегий).

Криптография – шифрование информации с помощью специальных алгоритмов (например, шифрование данных при их пересылке по Интернету; или использование электронной цифровой подписи).

Противодействие атакам вредоносных программ (англ. «malware») – предполагает использование внешних накопителей информации только от проверенных источников, антивирусных программ, брандмауэров, регулярное выполнение резервного копирования важных данных и т.д. (вредоносных программ очень много и они делятся на ряд классов: вирусы, эксплойты, логические бомбы, трояны, сетевые черви и т.п.).

Регламентация – создание условий по обработке, передаче и хранению информации, в наибольшей степени обеспечивающих ее защиту (специальные нормы и стандарты для персонала по работе с информацией, например, предписывающие в определенные числа делать резервную копию электронной документации, запрещающие использование собственных флеш-накопителей и т.д.).

Принуждение – установление правил по работе с информацией, нарушение которых карается материальной, административной или даже уголовной ответственностью (штрафы, закон «О коммерческой тайне» и т.п.).

Побуждение – призыв к персоналу не нарушать установленные порядки по работе с информацией, т.к. это противоречит сложившимся моральным и этическим нормам (например, Кодекс профессионального поведения членов «Ассоциации пользователей ЭВМ США»).

Средства защиты информации:

Технические (аппаратные) средства – сигнализация, решетки на окнах, генераторы помех воспрепятствования передаче данных по радиоканалам, электронные ключи и т.д.

Программные средства – программы-шифровальщики данных, антивирусы, системы аутентификации пользователей и т.п.

Смешанные средства – комбинация аппаратных и программных средств. Организационные средства – правила работы, регламенты, законодательные акты в сфере защиты информации, подготовка помещений с компьютерной техникой и прокладка сетевых кабелей с учетом требований по ограничению доступа к информации и пр.

Использование межсетевых экранов для защиты локальных сетей

Одним из самых популярных методов защиты локальной сети от атак извне является использование межсетевого экрана (МСЭ). Межсетевой экран(firewall,

брандмауэр) представляет собой программную или программно-аппаратную систему, которая устанавливается на границе охраняемой вычислительной сети и осуществляет фильтрацию сетевого трафика в обе стороны, разрешая или запрещая прохождение определенных пакетов внутрь локальной сети (в периметр безопасности) или из нее в зависимости от выбранной политики безопасности. Однако, только фильтрацией пакетов задачи современных МСЭ не ограничиваются, они выполняют также множество дополнительных действий:

- кэширование информации, когда часть полученной из внешней сети информации временно сохраняется на локальных запоминающих устройствах, что позволяет экономить время и потребляемый трафик при повторном обращении к той же информации;
- трансляция адресов, позволяющая использовать для внешних коммуникаций компьютеров локальной сети только один IP-адрес – адрес самого брандмауэра, внутренние адреса локальной сети могут быть любыми;
- переадресация, позволяющая отправлять пакеты, приходящие на некоторый IP-адрес, на компьютер с другим адресом. Это позволяет, например, распределить нагрузку на Web-сервер.

Существуют два основных типа МСЭ: пакетные фильтры и шлюзы приложений. При этом оба типа могут быть реализованы одновременно в одном брандмауэре. Пакетные фильтры (packet filter) представляют собой сетевые маршрутизаторы, которые принимают решение о том, пропускать или блокироватьпакет на основании информации в его заголовке (рис.55)

Пакетные фильтры работают с информацией в заголовках IP, ICMP, TCP и UDP- пакетов. Правила фильтрации пакетов задаются на основе следующих данных:

- название сетевого интерфейса и направление передачи информации;
- IP-адреса отправителя и получателя;
- протокол более высокого уровня (используется TCP или UDP);
- порт отправителя и получателя для протоколов TCP и UDP;
- опции IP (например, блокировка маршрутизации от источника);
- тип сообщения ICMP.

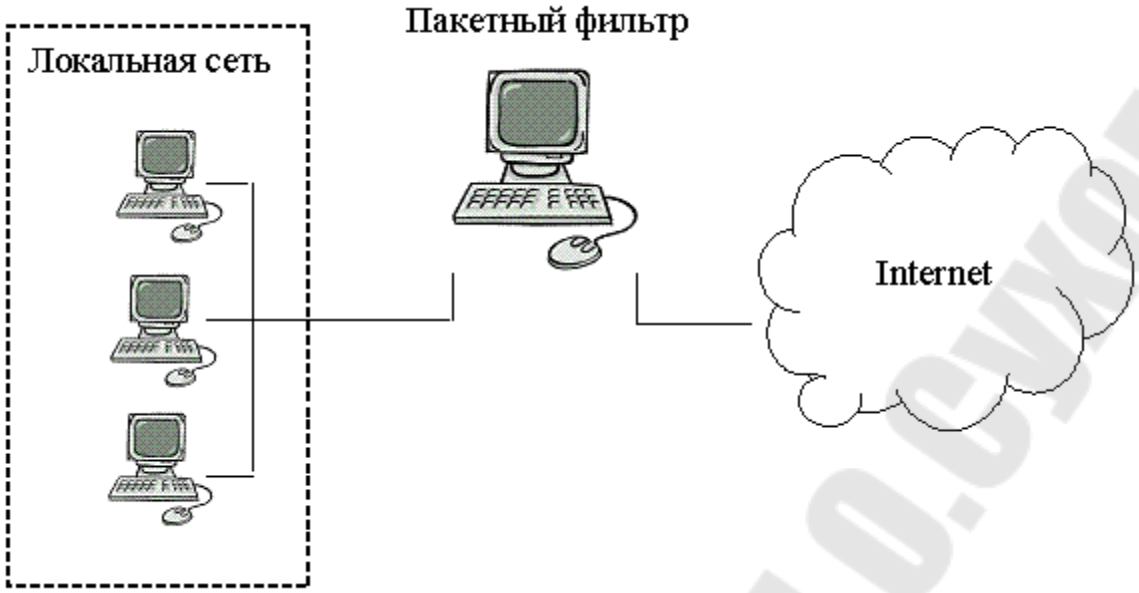


Рисунок 55 - Пакетный фильтр на границе локальной сети

При определении правил фильтрации необходимо придерживаться одной из двух стратегий политики безопасности:

1. Разрешить весь трафик, не запрещенный правилами фильтрации.

2. Запретить весь трафик, не разрешенный правилами фильтрации.

С точки зрения безопасности более предпочтительной является вторая стратегия, согласно которой задаются правила, разрешающие прохождение пакетов определенного типа, прохождение остальных пакетов запрещается. Это связано с тем, что, во-первых, количество запрещенных пакетов обычно гораздо больше, чем количество разрешенных, а во-вторых, со временем могут появиться новые службы, для которых при использовании первой стратегии необходимо будет дописывать запрещающие правила (если доступ к ним, конечно, нежелателен), в то время как вторая стратегия запретит доступ к ним автоматически.

Пакетные фильтры классифицируются на фильтры без памяти и фильтры с памятью (динамические). Первые из них фильтруют информацию только исходя из информации в заголовке рассматриваемого пакета. Динамические же пакеты учитывают при фильтрации текущее состояние соединений, формируя таблицы входящих и исходящих пакетов, и принимают решение на основании информации в нескольких взаимосвязанных пакетах.

Кроме того, пакетные фильтры могут реализовывать также множество дополнительных возможностей. Например, перенаправление пакетов, дублирование пакетов, подсчет трафика, ограничение полосы пропускания, запись пакетов в файл протокола и многое другое. Настройка пакетного фильтра требует от администратора значительной квалификации и понимания принципов работы всех протоколов стека TCP/IP от протоколов прикладного уровня до протоколов сетевого уровня.

Большинство современных операционных систем имеют встроенные пакетные фильтры, например ipfw в Unix или Internet Connection Firewall в Windows. Функции пакетного фильтра могут выполнять и аппаратные маршрутизаторы, например, CISCO PIX 515E.

Главным недостатком пакетных фильтров является невозможность осуществления фильтрации пакетов по содержимому информационной части пакетов, то есть по данным, относящимся к пакетам более высокого уровня. Этот недостаток может быть устранен путем использования шлюзов приложений (application gateway, proxy-server). Proxy-серверы работают на прикладном уровне, обеспечивая работу той или иной сетевой службы. При этом в отличие от пакетных фильтров, которые лишь перенаправляют пакет из одной сети в другую, proxy-серверы принимают запрос от клиента и направляют его во внешнюю сеть от своего имени, разрывая таким образом нормальный сетевой трафик (см. рис.56). Поэтому брандмауэр в виде шлюза приложений может быть реализован на компьютере всего с одним сетевым интерфейсом.

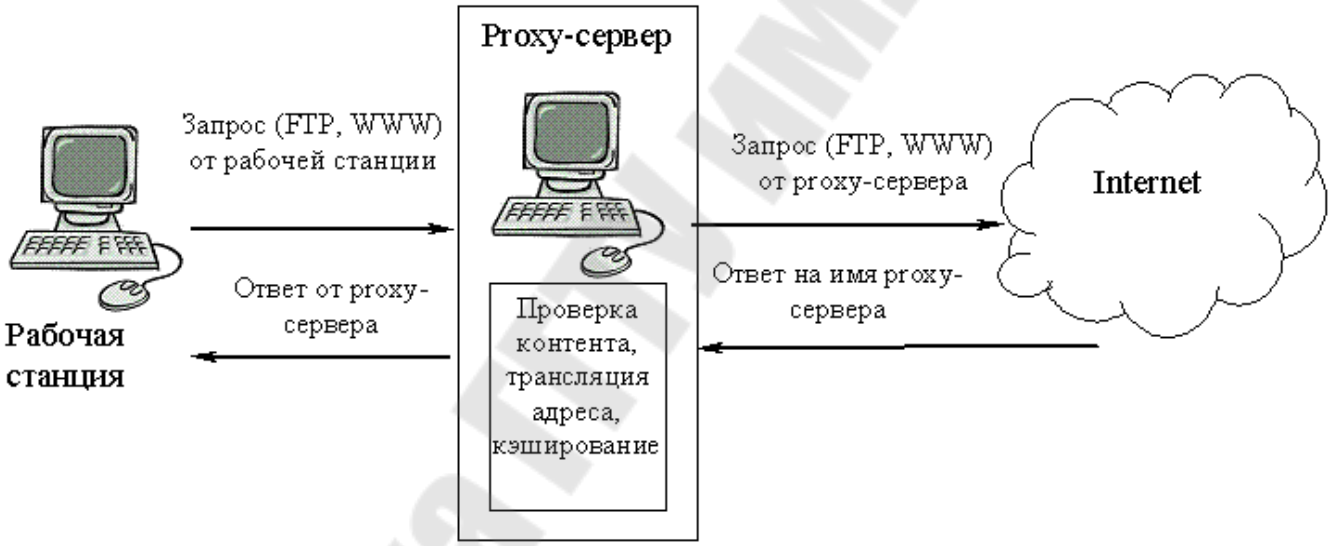


Рисунок 56 – Выполонение запроса через proxy сервер

Поясним схему на рис.56. Клиент формирует запрос на сервер какой-либо службы в Internet (например, запрос на внешний Web-сервер). Запрос поступает на proxy-сервер (конфигурация брандмауэра должна быть такова, чтобы все запросы к какой-либо службе в Internet обязательно поступали на соответствующий proxy-сервер). Приняв запрос от клиента, proxy-сервер проверяет его по заданным правилам фильтрации содержимого пакета и, если запрос не содержит запрещенных параметров, формирует пакет с запросом уже от своего имени (со своим обратным адресом) внешнему серверу. Ответ от внешнего сервера поступает, очевидно, на имя proxy-сервера. Пройдя проверку, аналогичную запросу, ответ может быть принят либо отвергнут. Если ответ принят -ответ направляется на адрес клиента, первоначально сформировавшего запрос.

Фильтрация содержимого может осуществляться по множеству параметров:

– IP-адрес отправителя и получателя;
– запрашиваемый URL;
– наличие вложений (приложения Java, компоненты ActiveX) и т.п.
– время запроса и другие, в зависимости от используемого proxy-сервера.

Важной функцией современных proxy-серверов является трансляция сетевых адресов (Network Address Translation, NAT), которая подразумевает замену адреса клиента в запросе во внешнюю сеть на собственный адрес (или несколько адресов) proxy-сервера. Это позволяет скрыть от посторонних структуру внутренней сети, список используемых в ней адресов. С другой стороны это позволяет иметь на всю локальную сеть лишь один легальный IP-адрес, который должен быть присвоен proxy-серверу. Рабочие станции внутри сети могут иметь любые IP-адреса, в том числе и те, которые запрещено использовать во внешней сети. NAT может быть организована по статической и динамической схеме. При статической трансляции адрес клиента в локальной сети привязывается к конкретному адресу, который транслируется во внешнюю сеть. Динамическая трансляция предполагает наличие диапазона доступных внешних адресов и при каждом запросе клиента proxy-сервер выделяет один из свободных адресов для представления клиента во внешней сети, по окончании транзакции этот адрес возвращается в список свободных и может быть использован в дальнейшем для передачи запроса другого клиента. Развитием идеи NAT стала трансляция адресов портов (Network Address Port Translation, NAPT), когда один и тот же IP-адрес распределяется при трансляции на несколько пользователей и каждому пользователю сопоставляется в отправляемом во внешнюю сеть пакете уникальная комбинация IP-адреса и номера порта отправителя. Иными словами, различным пользователям сети proxy-сервер сопоставляет один и тот же IP-адрес, но присваивает различные номера портов в исходящих запросах. Это стало возможным за счет того, что порт отправителя зачастую не несет в запросе никакой полезной информации и может быть использован для уникальной идентификации клиента локальной сети для proxy-сервера.

Современные proxy-серверы выполняют обычно еще одну важную функцию – кэширование информации. Информация, пришедшая на proxy-сервер, сохраняется на локальных запоминающих устройствах, и при очередном запросе клиента запрашиваемая им информация сначала ищется в локальной памяти, и только если ее там нет - запрос передается во внешнюю сеть. Это позволяет уменьшить объем трафика, потребляемого из внешней сети, а также уменьшить время доступа к информации для конечного клиента.

Рассмотрим свойства наиболее распространенных proxy-серверов.

Microsoft Proxy Server (версия 2.0) представляет собой брандмауэр с расширяемым набором функций и сервер кэширования информации, обеспечивает поддержку протоколов HTTP и gopher, а также поддержку клиентских приложений (например, Telnet и RealAudio) для компьютеров интрасети, использующих

протоколы TCP/IP или IPX/SPX, поддерживает VPN, выполняет функции фильтра пакетов. Работает в среде Windows.

Squid - высокопроизводительный кэширующий proxy-сервер для web-клиентов с поддержкой протоколов FTP, gopher и HTTP, имеющий реализации как под Unix, так и под Windows- платформы.Squid хранит метаданные и особенно часто запрашиваемые объекты в ОЗУ, кэширует DNS-запросы, поддерживает неблокирующие DNS-запросы и реализует негативное кэширование неудачных запросов. Поддерживает протокол ICP (Internet Casing Protocol), позволяющий организовывать нескольким серверам иерархические структуры кэширования.

Помимо перечисленных, на практике могут быть использованы так называемые персональные межсетевые экраны. Они устанавливаются на компьютер пользователя, и все правила безопасности задаются для обмена этого компьютера с внешней сетью. Это позволяет настроить политику безопасности персонально под каждого пользователя непосредственно на его рабочей станции. Примером подобного МСЭ является брандмауэр AtGuard, который включает в себя функции proxy-сервера и локального пакетного фильтра. AtGuard способен блокировать баннеры, файлы cookie, Java -скрипты и апплеты, а также элементы ActiveX. Еще одна особенность AtGuard – способность работать в режиме обучения, когда при каждой попытке подключиться к какому-либо порту запрашивается разрешение на установку соединения, и сделанный пользователем выбор становится правилом для дальнейшей работы программы.

Используемые на практике МСЭ представляют собой интегрированную систему защиты, включающую и пакетный фильтр, и proxy-сервер. Они могут располагаться как на одном, так и на нескольких компьютерах, в связи с чем существует возможность выбора архитектуры используемого МСЭ [8].

Архитектура с использованием в качестве МСЭ компьютера с двумя сетевыми интерфейсами похожа на схему подключения пакетного фильтра (рис. 57), но на МСЭ должна быть отключена возможность маршрутизации пакетов. Это позволяет полностью блокировать трафик во внешнюю сеть на этом компьютере, а все необходимые сервисы должны обеспечиваться proxy-серверами, работающими на двухканальном компьютере. Для обеспечения дополнительной защиты можно поместить маршрутизатор с фильтрацией пакетов между внешней сетью и двухканальным компьютером. Архитектура с экранированным узлом предполагает использование одновременно и пакетного фильтра, и proxy-сервера (рис.5.5).

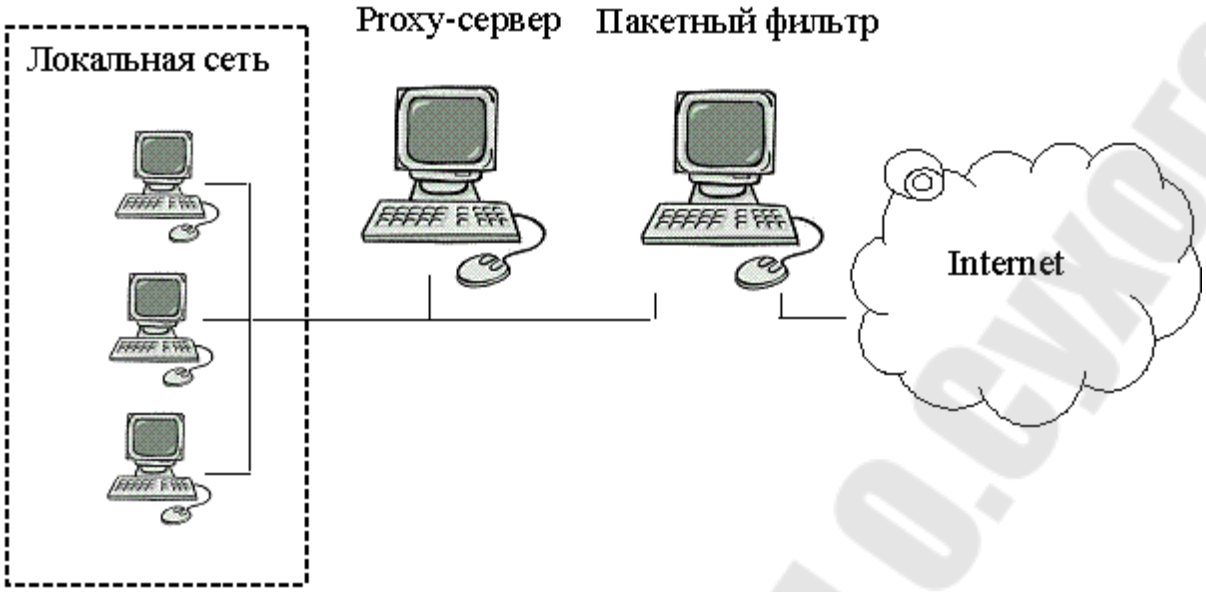


Рисунок 57 – МСЭ с использованием экранированного узла

На границе с внешней сетью устанавливается пакетный фильтр, который должен блокировать потенциально опасные пакеты, чтобы они не достигли прикладного шлюза (proxy-сервера) и локальной сети. Он отвергает или пропускает трафик в соответствии со следующими правилами:

- трафик из внешней сети к прикладному шлюзу пропускается;
- прочий трафик из внешней сети блокируется;
- пакетный фильтр блокирует любой трафик из локальной сети во внешнюю, если он не идет от прикладного шлюза.

Прикладной шлюз должен обеспечивать функции proxy-сервера для всех потенциально опасных служб и для работы ему достаточно одного сетевого интерфейса. Подобная схема подключения брандмауэра отличается большей гибкостью по сравнению с двухканальным МСЭ, поскольку пакетный фильтр может позволить пропустить запросы к надежным сервисам в обход прикладного шлюза. Этими надежными сервисами могут быть те сервисы, для которых нет proxy-сервера, и которым можно доверять в том смысле, что риск использования этих сервисов считается приемлемым.

Развитием концепции изолированного узла стала архитектура с изолированной подсетью. Здесь используется два пакетных фильтра (рис.58)для организации изолированной подсети, которую еще называют демилитаризованной зоной (DMZ).

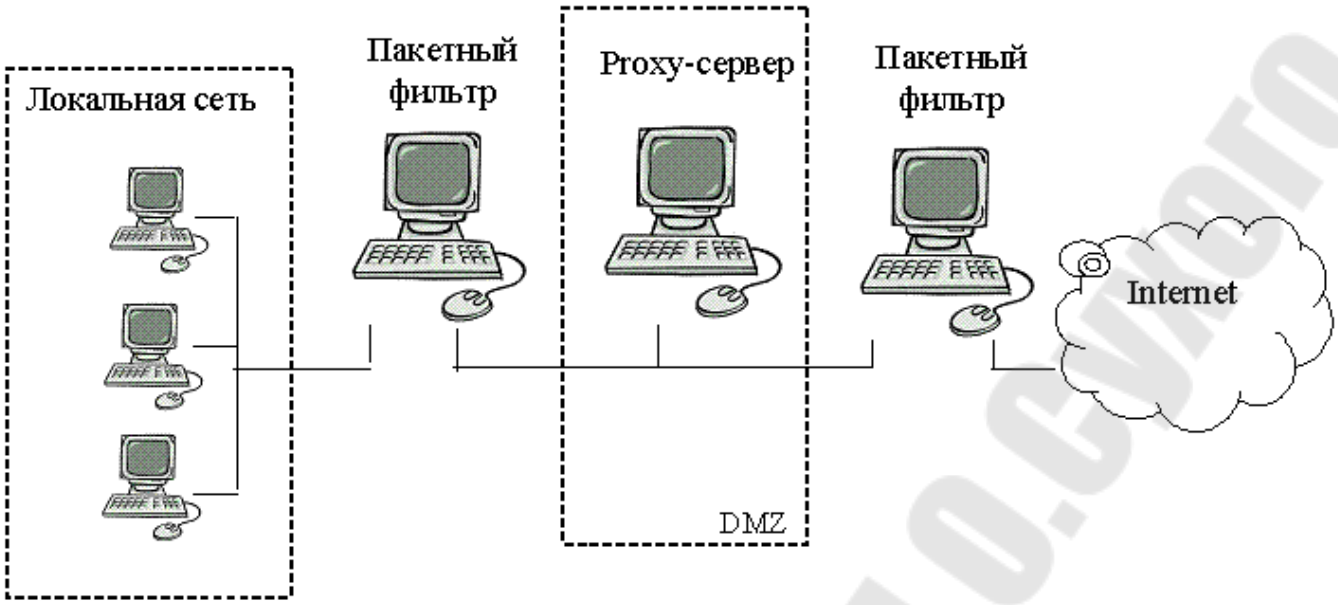


Рисунок 58 – МСЭ с использованием экранированной подсети

Внутри изолированной подсети должен находиться прикладной шлюз, а также могут находиться различные информационные серверы (mail, WWW, FTP), модемные пулы и т.п. Пакетный фильтр, установленный на границе с внешней сетью, должен фильтровать пакеты по следующим правилам:

– пропускать прикладной трафик от прикладного шлюза во внешнюю сеть;
– пропускать прикладной трафик из внешней сети к прикладному шлюзу;
– все остальные виды трафика блокировать.

Внутренний пакетный фильтр управляет трафиком «локальная сеть – демилитаризованная зона» согласно следующим правилам:

– трафик от прикладного шлюза к внутренним системам пропускается;
– трафик к прикладному шлюзу от внутренних систем пропускается;
– трафик к информационным серверам внутри DMZ пропускается;
– все остальные виды трафика блокировать.

Такая схема дает возможность еще более гибко формировать политику безопасности, задавая различные правила фильтрации для двух пакетных фильтров (например, можно разрешить прохождения FTP-пакетов в DMZ из локальной сети для обновления информации на WWW-сервере и запретить доступ по FTP-протоколу к DMZ из внешней сети). Компьютеры, расположенные в демилитаризованной зоне, подвержены большему количеству атак, чем компьютеры локальной сети. Поэтому все компьютеры, находящиеся в DMZ, должны быть максимально укреплены (bastion host, укрепленный компьютер). На них должны быть удалены все неиспользуемые службы, максимально активизированы средства безопасности операционных систем (ужесточаются права доступа к объектам, минимизируется количество зарегистрированных субъектов,ведется строгий аудит).

Очевидно, что можно создать несколько экранированных подсетей, отделенных друг от друга собственным пакетным фильтром с определением правил доступа для каждой из подсетей. Выбор конкретной архитектуры МСЭ зависит от стоящих перед администратором задач, условий функционирования, стоимости того или иного решения

Современные информационные системы функционируют в условиях постоянных угроз, исходящих из сети Интернет. Для защиты от этих угроз существует множество средств на различных уровнях. Для обеспечения конфиденциальности ицелостностипередаваемой по глобальной сети информации можно использовать защищенные сетевые протоколы, которые используют криптографические методы. Для предотвращениявторжения извне, защиты от атак типа DoS необходимо использовать межсетевые экраны, основная задача которых – фильтрация входящего и исходящего сетевого трафика в зависимости от принятой политики безопасности.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Основная литература

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Ученик для вузов. 4-е изд. – СПб.: Питер, 2015.
2. Таненбаум Э. Компьютерные сети. – М. «Вильямс», 2011

Дополнительная литература

3. Администрирование сети на основе Windows 2000. Учебный курс MCSE. Сертификационный экзамен 70-216. - СПб.: БХВ-Петербург, 2004
4. Вишневский В.М. Теоретические основы проектирования компьютерных сетей. – Москва: Техносфера, 2003.
5. Крелл М., Манн С. Linux. Администрирование сетей TCP/IP. – М. «Вильямс», 2003
6. Кузьменко Н. Компьютерные сети и сетевые технологии. – СПб.: «Наука и Техника», 2013
7. Попов И., Максимов Н. Компьютерные сети. – Москва: «Инфра-М», 2013
8. Стахнов А. Сетевое администрирование Linux. - СПб.: Питер-пресс, 2004
9. Microsoft Windows 2000: Server и Professional. Русские версии / А.Г. Андреев [и др.] Под общ. ред. А.Н. Чекмарева и Д.Б.Вишнякова. – СПб.: БХВ-Петербург, 2003