МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Государственное образовательное учреждение
высшего профессионального образования

“Оренбургский государственный университет”

Л.Ф. ГАЙСИНА

СЕТИ ЭВМ
И
ТЕЛЕКОММУНИКАЦИИ

Рекомендовано Ученым советом государственного образовательного учреждения высшего профессионального образования “Оренбургский государственный университет” в качестве учебного пособия для студентов, обучающихся по программам высшего профессионального образования по специальности “Программное обеспечение вычислительной техники и автоматизированных систем”

Оренбург 2004

ББК 32.973.202я73

Г 14
УДК 004.7(075)

Рецензент
кандидат технических наук, доцент Раимова А.Т.

Гайсина Л.Ф.
Г - 14 **Сети ЭВМ и телекоммуникации: Учебное пособие. - Оренбург: ГОУ ОГУ, 2004. - 160 с.**

ISBN _____

Учебное пособие посвящено архитектуре вычислительных сетей: рассматривается классификация вычислительных сетей, сетевые топологии и методы доступа к среде передачи данных, эталонная модель взаимодействия открытых систем. Приведены сведения об устройствах объединения сетей: концентраторах, мостах, коммутаторах и маршрутизаторах. Приводится классификация сетевых протоколов и рассматриваются стандартные протоколы. Особое внимание уделяется протоколам Internet сетевого и транспортного уровней. Рассмотрены различные информационные сервисы Internet, служба WWW, система доменной адресации, почта и т.д. Раскрыты основы языка гипертекстовой разметки HTML.

Учебное пособие предназначено для студентов, обучающихся по программам высшего образования по специальности 220400, при изучении дисциплины "Сети ЭВМ и телекоммуникации".

Г 2404040000 ББК 32.973.202я73

ISBN _____

Введение

Сложный характер и динамизм современных мирохозяйственных связей привел к необходимости создания новых телекоммуникационных технологий, порождающих новые услуги и соответственно увеличивающуюся потребность в них.

Объем и способы информирования специалистов с помощью средств компьютерных коммуникаций коренным образом изменились за последние год-два. И если ранее подобные средства предназначались лишь для узкого круга специалистов и опытных пользователей, то теперь они рассчитаны на самую широкую аудиторию.

В настоящее время передача данных с помощью компьютеров, использование локальных и глобальных компьютерных сетей становится столь же распространенным, как и сами компьютеры.

Целью данного пособия является подготовка студентов к умелому использованию локальных и глобальных компьютерных сетей, коммуникационного оборудования и программных средств.

После окончания изучения представленного раздела студенты должны уметь ориентироваться среди богатого разнообразия предлагаемой аппаратуры и программ, в принципах функционирования локальных вычислительных сетей, в сетевых протоколах и основах Internet-технологий.

Изложение материала во всех главах опирается на множество примеров. После каждой главы представлены контрольные вопросы для закрепления изученного теоретического материала, тесты для самоконтроля, а также список рекомендуемой литературы для изучения курса.

В заключение, автор выражает признательность рецензенту и научному редактору за внимательное прочтение рукописи и замечания, способствовавшие улучшению качества предлагаемого пособия.

1 Общие сведения о вычислительных сетях

1.1 Назначение вычислительных сетей

Вычислительные сети (ВС) появились давно. Еще на заре появления компьютеров существовали огромные системы, известные как системы разделения времени. Они позволяли использовать центральную ЭВМ с помощью удаленных терминалов. Такой терминал состоял из дисплея и клавиатуры. Внешне выглядел как обычный персональный компьютер, но не имел собственного процессорного блока. Пользуясь такими терминалами, сотни, а иногда тысячи сотрудников имели доступ к центральной ЭВМ.

Такой режим обеспечивался благодаря тому, что система разделения времени разбивала время работы центральной ЭВМ на короткие интервалы времени, распределяя их между пользователями. При этом создавалась иллюзия одновременного использования центральной ЭВМ многими сотрудниками.

В 70-х годах большие ЭВМ уступили место миникомпьютерным системам, использующим тот же режим разделения времени. Но технология развивалась, и с конца 70-х годов на рабочих местах появились персональные компьютеры. Однако, автономно работающие персональные компьютеры не дают непосредственного доступа к данным всей организации и не позволяют совместно использовать программы и оборудование.

С этого момента начинается современное развитие компьютерных сетей.

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

Самая простая сеть (network) состоит из нескольких персональных компьютеров, соединенных между собой сетевым кабелем (рисунок 1). При этом в каждом компьютере устанавливается специальная плата сетевого адаптера (NIC), осуществляющая связь между системной шиной компьютера и сетевым кабелем /1/.

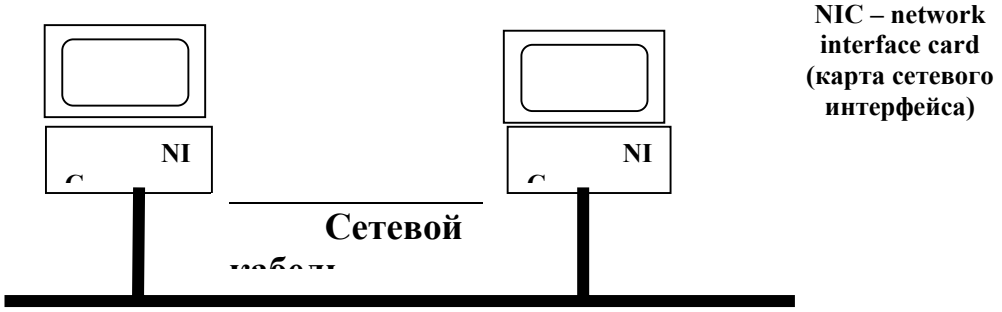


Рисунок 1 - Структура простейшей вычислительной сети

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

за ее пределами (рисунок 2).

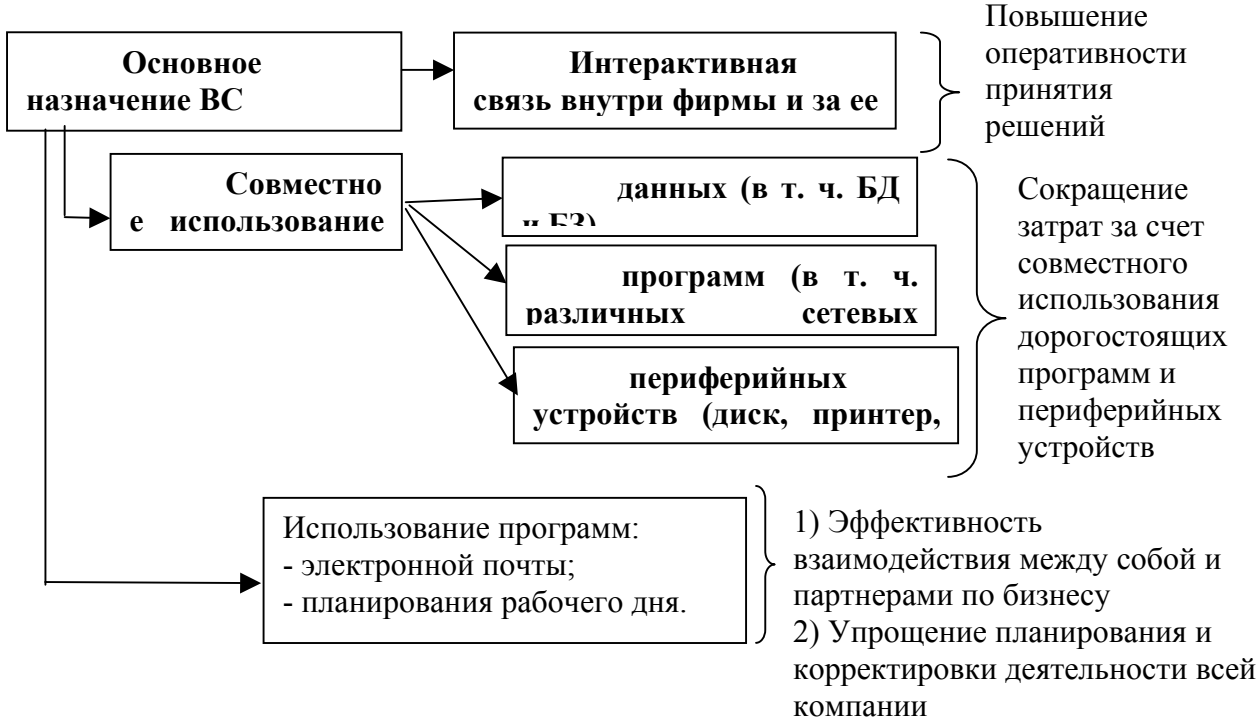


Рисунок 2 - Назначение вычислительной сети

Ресурсы – представляют собой данные (в том числе корпоративные базы данных и знаний), приложения (в том числе различные сетевые программы), а также периферийные устройства, такие как принтер, сканер, модем и т. д.

До объединения персонального компьютера в сеть каждый пользователь должен был иметь свой принтер, плоттер и другие периферийные устройства, а также на каждом из компьютеров должны были быть установлены одни и те же программные средства, используемые группой пользователей.

Другой привлекательной стороной сети является наличие программ электронной почты и планирования рабочего дня. Благодаря им, сотрудники эффективно взаимодействуют между собой и партнерами по бизнесу, а планирование и корректировка деятельности всей компании осуществляется значительно проще. Использование компьютерных сетей позволяет: а) повысить эффективность работы персонала фирмы; б) снизить затраты за счет совместного использования данных, дорогостоящих периферийных устройтв и программных средств (приложений).

Основными характеристиками вычислительной сети являются:

- операционные возможности сети;
- временные характеристики;
- надежность;
- производительность;
- стоимость.

Операционные возможности сети характеризуются такими условиями, как:

- предоставление доступа к прикладным программным средствам, БД,

БЗ, и т. д.;

- удаленный ввод заданий;
- передача файлов между узлами сети;
- доступы к удаленным файлам;
- выдача справок об информационных и программных ресурсах;
- распределенная обработка данных на нескольких ЭВМ и т. д.

Временные характеристики сети определяют продолжительность обслуживания запросов пользователей:

- среднее время доступа, которое зависит от размеров сети, удаленности пользователей, загрузки и пропускной способности каналов связи и т. д.;
- среднее время обслуживания.

Надежностные характеризуют надежность, как отдельных элементов сети, так и сеть в целом.

Пакет как основная единица информации в вычислительных сетях. При обмене данными между персональными компьютерами любое информационное сообщение разбивается программами передачи данных на небольшие блоки данных, которые называются *пакетами* (рисунок 3).

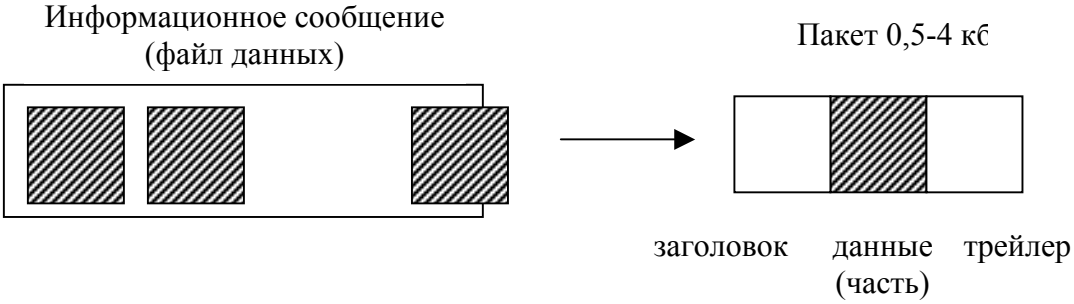


Рисунок 3 - Информационное сообщение

Связано это с тем, что данные обычно содержатся в больших по размерам файлах, и если передающий компьютер пошлет его целиком, то он надолго заполнит канал связи и «свяжет» работу всей сети, то есть будет препятствовать взаимодействию других участников сети. Кроме этого, возникновение ошибок при передаче крупных блоков вызовет большие затраты времени, чем на его повторную передачу.

Пакет – основная единица информации в компьютерных сетях. При разбиении данных на пакеты скорость их передачи возрастает на столько, что каждый компьютер сети получает возможность принимать и передавать данные практически одновременно с остальными ПК /2/.

При разбиении данных на пакеты сетевая операционная система к собственно передаваемым данным добавляет специальную добавляющую информацию:

- заголовок, в котором указывается адрес отправителя, а также информация по сбору блоков данных в исходное информационное сообщение при их приеме получателем;
- трейлер, в котором содержится информация для проверки безошибочности в передаче пакета. При обнаружении ошибки передача пакета

должна повториться.

Способы организации передачи данных между персональными компьютерами. Передача данных между компьютерами и прочими устройствами происходит параллельно или последовательно.

Так большинство персональных компьютеров пользуется параллельным портом для работы с принтером. Термин «параллельно» означает, что данные передаются одновременно по нескольким проводам.

Чтобы послать байт данных по параллельному соединению, компьютер одновременно устанавливает весь байт на восьми проводах. Схему параллельного соединения можно иллюстрировать на рисунке 4.

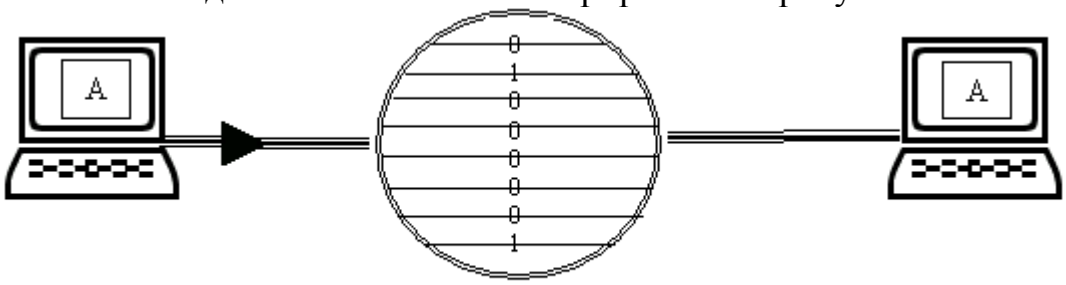


Рисунок 4 – Параллельное соединение

Как видно из рисунка, параллельное соединение по восьми проводам позволяет передать байт данных одновременно.

Напротив, последовательное соединение подразумевает передачу данных по очереди, бит за битом. В сетях чаще всего используется именно такой способ работы, когда биты выстраиваются друг за другом и последовательно передаются (и принимаются тоже), что иллюстрирует рисунок 5.

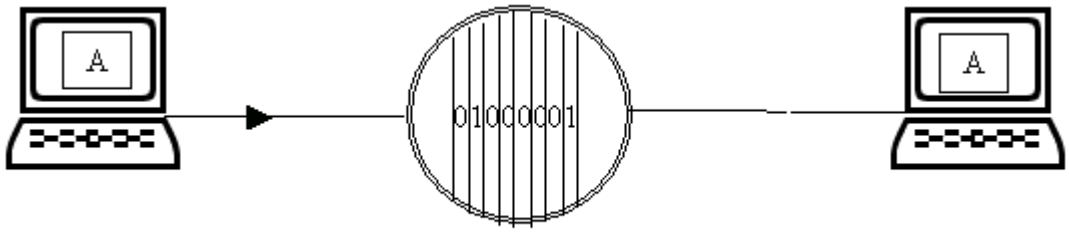


Рисунок 5 – Последовательное соединение

При соединении по сетевым каналам используют три различных метода. Соединение бывает: симплексное, полудуплексное и дуплексное.

О *симплексном соединении* говорят, когда данные перемещаются только в одном направлении (рисунок 6). *Полудуплексное соединение* позволяет данным перемещаться в обоих направлениях, но в разное время.

И, наконец, *дуплексное соединение* позволяет данным перемещаться в обоих направлениях одновременно.

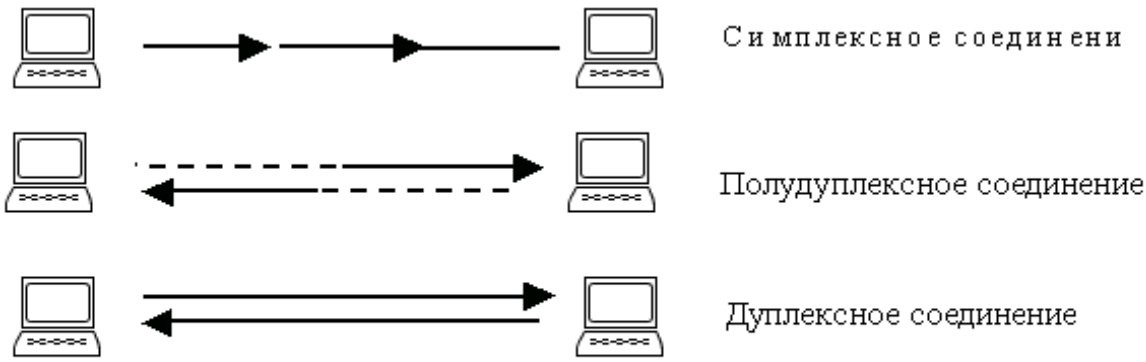


Рисунок 6 – Типы соединений

1.2 Архитектура "клиент-сервер"

Сеть ЭВМ (компьютерная сеть, или вычислительная сеть - ВС) - это совокупность компьютеров и терминалов, соединенных с помощью каналов связи в единую систему, удовлетворяющую требованиям распределенной обработки данных, совместного использования общих информационных и вычислительных ресурсов /2/.

Часто возникает путаница между распределенными системами и сетями ЭВМ. Работая с распределенной системой, пользователь может не иметь ни малейшего представления на каких процессорах, где, с использованием конкретно каких физических ресурсов будет исполняться его программа. В сети, поскольку все машины там автономны, пользователь должен делать все явно. Основное различие между этими системами лежит в организации их программного обеспечения. И там и там происходит передача информации. В сети - пользователь, в распределенной системе - система.

Распределенные вычисления в компьютерных сетях основаны на архитектуре "клиент-сервер", ставшей доминирующим способом обработки данных. Термины "клиент" и "сервер" обозначают роли, которые играют различные компоненты в распределенной среде вычислений. Компоненты "клиент" и "сервер" не обязательно должны работать на разных машинах, хотя обычно это так и есть - клиент-приложение находится на рабочей станции пользователя, а сервер - на специальной выделенной машине. Наиболее распространены слеующие виды серверов: файл-серверы, серверы баз данных, серверы печати, серверы электронной почты, Web-сервер и другие. В последнее время интенсивно внедряются многофункциональные серверы приложений.

Клиент формирует запрос на сервер для выполнения соответствующих функций. Например, файл-сервер обеспечивает хранение данных общего пользования, организует доступ к ним и передает данные клиенту. Обработка данных распределяется в том или ином соотношении между сервером и клиентом. В последнее время долю обработки, приходящуюся на клиента, стали называть "толщиной" клиента.

Развитие архитектуры "клиент-сервер" происходит по спирали и в

настоящее время намечается тенденция централизации вычислений, то есть замены "толстых" клиентов - рабочих станций на основе высокопроизводительных ПЭВМ, оснащенных мощным программным обеспечением для поддержки прикладных программ, мультимедийных средств, навигационного и графического интерфейса - "тонкими" клиентами. Характерный пример "тонкого" клиента - архитектура Sun Ray Hot Desk, предложенная компанией Sun Microsystems.

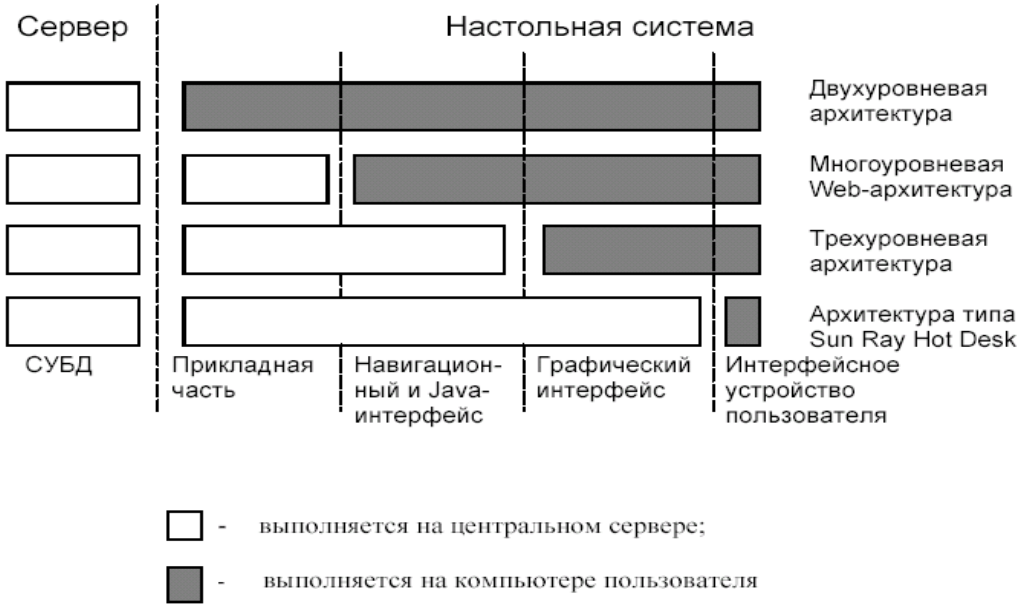


Рисунок 7 - Ранжирование клиентов по "толщине"

Архитектура Sun Ray Hot Desk предполагает использование настольных систем типа графических терминалов Sun Ray 1, имеющих минимум программных и аппаратных средств, но обладающих широкими возможностями работы с приложениями в соответствии с основной идеей "тонких" клиентов - вынести на сервер все, вплоть до виртуальных драйверов устройств, включая драйвер монитора. Историческими предшественниками "тонких" клиентов были алфавитно-цифровые терминалы, подключавшиеся к главным ЭВМ, или мэйнфреймам (mainframe) через специализированные интерфейсы или универсальные последовательные порты.

Мэйнфреймы - классический пример централизации вычислений, поскольку в едином комплексе были сконцентрированы все вычислительные ресурсы, хранение и обработка огромных массивов данных. Основные достоинства централизованной архитектуры - простота администрирования и защиты информации. Все терминалы были однотипными - следовательно, устройства на рабочих местах пользователей вели себя предсказуемо и в любой момент могли бы быть заменены, затраты на обслуживание терминалов и линий связи также легко прогнозировались /3/.

Революция, вызванная появлением персональных компьютеров, сделала возможным иметь вычислительные и информационные ресурсы на рабочем столе пользователя и управлять ими по собственному разумению с помощью цветного оконного графического интерфейса. Увеличение производительности

ПК позволило перенести части системы (интерфейс с пользователем, прикладную логику) для выполнения на персональном компьютере, непосредственно на рабочем месте, а функции обработки данных оставить на центральном компьютере. Система стала распределенной - одна часть функций выполняется на центральном компьютере, другая - на персональном, который связан с центральным посредством коммуникационной сети. Таким образом, появилась клиент-серверная модель взаимодействия компьютеров и программ в сети и на этой основе стали развиваться средства разработки приложений для реализации информационных систем.

Однако двухуровневая архитектура "клиент-сервер" (рисунок 7) имеет такие существенные недостатки, как сложность администрирования и низкая информационная безопасность, особенно заметные при сравнении ее с централизованной архитектурой мэйнфреймов (таблица 1).

Таблица 1 - Сравнение централизованной архитектуры мэйнфреймов и двухуровневой архитектуры "клиент-сервер"

Централизованная архитектура мэйнфреймов	Двухуровневая архитектура "клиент-сервер"
Вся информационная система на центральном компьютере	Систему, состоящую из большого числа разнотипных компьютеров, на которых работают разнородные приложения, трудно администрировать
На рабочих местах простые устройства доступа, дающие возможность пользователю управлять процессами в информационной системе	Компьютеры сложны в конфигурировании и поиске неисправностей, стоимость обслуживания достаточно высока
Устройство доступа общается с центральным компьютером посредством простого, аппаратно реализованного протокола	Компьютер весьма уязвим для вирусов и несанкционированного доступа

1.3 Классификация вычислительных сетей

На сегодня нет общепризнанной классификации сетей. Есть два общепризнанных фактора для их различения: **технология передачи** и **масштаб**.

Есть два основных типа технологий передачи, используемые в сетях:

- вещание (от одного ко многим);
- точка-точка.

Сети типа вещание имеют единый канал передачи данных, который используют все машины сети. Пакет, отправленный какой-то машиной, получают все другие машины сети. В определенном поле пакета указан адрес получателя. Каждая машина проверяет это поле и если она обнаруживает в этом поле свой адрес, она приступает к обработке этого пакета; если в этом поле не ее адрес, то она просто игнорирует этот пакет.

Сети типа вещание, как правило, имеют режим, когда один пакет адресуется всем машинам в сети. Это, так называемый, режим широкого вещания. Есть в таких сетях режим группового вещания - один и тот же пакет получают машины принадлежащие к определенной группе в сети.

Сети точка-точка соединяют каждую пару машин индивидуальным каналом. Поэтому, прежде чем пакет достигнет адресата, он проходит через несколько промежуточных машин. В этих сетях возникает потребность в маршрутизации. От ее эффективности зависит скорость доставки сообщений, распределение нагрузки в сети.

Сети типа вещание, как правило, используются на географически небольших территориях. Сети точка-точка - для построения крупных сетей, охватывающих большие регионы /4/.

Масштаб сети - другой критерий для классификации сетей. Протяженность связи, которую обеспечивает вычислительная сеть, может быть различной: в пределах одного помещения, здания, предприятия, региона, континента или всего мира.

К *локальным сетям - Local Area Networks (LAN)* - относят сети компьютеров, сосредоточенные на небольшой территории (обычно в радиусе не более 1-2 км). В общем случае локальная сеть представляет собой коммуникационную систему, принадлежащую одной организации. Из-за коротких расстояний в локальных сетях имеется возможность использования относительно дорогих высококачественных линий связи, которые позволяют, применяя простые методы передачи данных, достигать высоких скоростей обмена данными порядка 100 Мбит/с. В связи с этим услуги, предоставляемые локальными сетями, отличаются широким разнообразием и обычно предусматривают реализацию в режиме on-line.

Глобальные сети - Wide Area Networks (WAN) - объединяют территориально рассредоточенные компьютеры, которые могут находиться в различных городах и странах. Так как прокладка высококачественных линий связи на большие расстояния обходится очень дорого, в глобальных сетях часто используются уже существующие линии связи, изначально предназначенные совсем для других целей. Например, многие глобальные сети строятся на основе телефонных и телеграфных каналов общего назначения. Из-за низких скоростей таких линий связи в глобальных сетях (десятки килобит в секунду) набор предоставляемых услуг обычно ограничивается передачей файлов, преимущественно не в оперативном, а в фоновом режиме, с использованием электронной почты. Для устойчивой передачи дискретных данных по некачественным линиям связи применяются методы и оборудование, существенно отличающиеся от методов и оборудования, характерных для

локальных сетей. Как правило, здесь применяются сложные процедуры контроля и восстановления данных, так как наиболее типичный режим передачи данных по территориальному каналу связи связан со значительными искажениями сигналов.

Городские сети (или сети мегаполисов) - Metropolitan Area Networks (MAN) - являются менее распространенным типом сетей. Эти сети появились сравнительно недавно. Они предназначены для обслуживания территории крупного города - мегаполиса. В то время как локальные сети наилучшим образом подходят для разделения ресурсов на коротких расстояниях и широковещательных передач, а глобальные сети обеспечивают работу на больших расстояниях, но с ограниченной скоростью и небогатым набором услуг, сети мегаполисов занимают некоторое промежуточное положение. Они используют цифровые магистральные линии связи, часто оптоволоконные, со скоростями от 45 Мбит/с, и предназначены для связи локальных сетей в масштабах города и соединения локальных сетей с глобальными. Эти сети первоначально были разработаны для передачи данных, но также они поддерживают и такие услуги, как видеоконференции и интегральную передачу голоса и текста. Развитие технологии сетей мегаполисов осуществлялось местными телефонными компаниями.

1.3.1 Локальные вычислительные сети

Локальные вычислительные сети (ЛВС) получили в настоящее время широкое распространение из-за небольшой сложности и невысокой стоимости. Они используются при автоматизации коммерческой, банковской деятельности, а также для создания распределенных, управляющих и информационно-справочных систем. ЛВС имеют модульную организацию (рисунок 8):

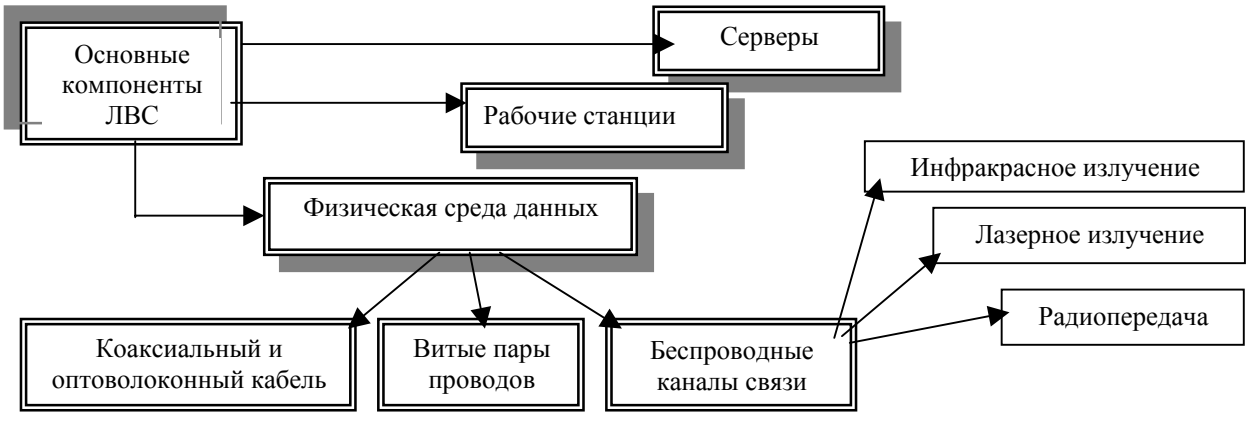


Рисунок 8 - Компоненты локальной вычислительной сети

Их основные компоненты - это

– серверы – это аппаратно-программные комплексы, которые исполняют функции управления распределением сетевых ресурсов общего доступа,

– рабочие станции – это компьютеры, осуществляющие доступ к сетевым ресурсам, предоставляемым сервером,

– физическая среда передачи данных (сетевой кабель)– это коаксиальные и оптоволоконные кабели, витые пары проводов, а также беспроводные каналы связи (инфракрасное излучение, лазеры, радиопередача).

Выделяется два основных типа локальных вычислительных сетей: одноранговые и на основе сервера. Различия между ними имеют принципиальное значение, так как определяют разные возможности этих сетей.

Одноранговые сети. В этих сетях все компьютеры равноправны: нет иерархии среди них; нет выделенного сервера. Как правило, каждый ПК функционирует и как рабочая станция (РС), и как сервер, то есть нет ПК ответственного за администрирование всей сети. Все пользователи решают сами, какие данные и ресурсы на своем компьютере сделать общедоступными по сети.

Рабочая группа – это небольшой коллектив, объединенный общей целью и интересами. Поэтому в одноранговых сетях чаще всего не более 10 компьютеров. Эти сети относительно просты, так как каждый ПК является одновременно и РС, и сервером. Нет необходимости в мощном центральном сервере или в других компонентах, обязательных для более сложных сетей.

В такие операционные системы, как MS Widows NT for Workstation, MS Widows 95/98, Widows 2000 встроена поддержка одноранговых сетей. Поэтому, чтобы установить одноранговую сеть, дополнительного программного обеспечения не требуется, а для объединения компьютеров применяется простая кабельная система.

Несмотря на то, что одноранговые сети вполне удовлетворяют потребностям небольших фирм, возникают ситуации, когда их использование является неуместным. В этих сетях защита предполагает установку пароля на разделяемый ресурс (например, каталог). Централизованно управлять защитой в одноранговой сети очень сложно, так как:

– пользователь устанавливает ее самостоятельно;

– «общие» ресурсы могут находиться на всех ПК, а не только на центральном сервере.

Такая ситуация – угроза для всей сети; кроме того, некоторые пользователи могут вообще не установить защиту. Если вопросы конфиденциальности являются для фирмы принципиальными, то такие сети применять не рекомендуется. Кроме того, так как в этих ЛВС каждый ПК работает и как РС, и как сервер, пользователи должны обладать достаточным уровнем знаний, чтобы работать и как пользователи, и как администраторы своего компьютера.

Сети на основе сервера. При подключении более 10 пользователей одноранговая сеть может оказаться недостаточно производительной. Поэтому большинство сетей используют выделенные серверы. *Выделенными* называются такие серверы, которые функционируют только как сервер (исключая функции РС или клиента). Они специально оптимизированы для быстрой обработки

запросов от сетевых клиентов и для управления защитой файлов и каталогов.

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

– Файл-серверы и принт-серверы. Они управляют доступом пользователей к файлам и принтерам.

– Серверы приложений (в том числе сервер баз данных, Web–сервер). На них выполняются прикладные части клиент серверных приложений (программ).

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

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

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

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

1.3.2 Сети отделов, кампусов, корпоративные сети

Еще одним популярным способом классификации сетей является их классификация по масштабу производственного подразделения, в пределах которого действует сеть. Различают сети отделов, сети кампусов и корпоративные сети.

Сети отделов - это сети, которые используются сравнительно небольшой группой сотрудников, работающих в одном отделе предприятия. Эти сотрудники решают некоторые общие задачи, например, ведут бухгалтерский учет или занимаются маркетингом. Считается, что отдел может насчитывать до 100-150 сотрудников.

Главной целью сети отдела является разделение локальных ресурсов, таких как приложения, данные, лазерные принтеры и модемы. Обычно сети отделов имеют один или два файловых сервера и не более тридцати пользователей. Сети отделов обычно не разделяются на подсети. В этих сетях локализуется большая часть трафика предприятия. Сети отделов обычно создаются на основе какой-либо одной сетевой технологии - Ethernet, Token Ring /5/.

Задачи управления сетью на уровне отдела относительно просты: добавление новых пользователей, устранение простых отказов, инсталляция новых узлов и установка новых версий программного обеспечения. Такой сетью может управлять сотрудник, посвящающий обязанностям

администратора только часть своего времени. Чаще всего администратор сети отдела не имеет специальной подготовки, но является тем человеком в отделе, который лучше всех разбирается в компьютерах, и само собой получается так, что он занимается администрированием сети.

Рисунок 9 - Пример сети масштаба отдела

Существует и другой тип сетей, близкий к сетям отделов, - *сети рабочих групп*. К таким сетям относят совсем небольшие сети, включающие до 10-20 компьютеров. Характеристики сетей рабочих групп практически не отличаются от описанных выше характеристик сетей отделов. Такие свойства, как простота сети и однородность, здесь проявляются в наибольшей степени, в то время как сети отделов могут приближаться в некоторых случаях к следующему по масштабу типу сетей - сетям кампусов.

Сети кампусов получили свое название от английского слова campus - студенческий городок. Именно на территории университетских городков часто возникала необходимость объединения нескольких мелких сетей в одну большую сеть. Сейчас это название не связывают со студенческими городками, а используют для обозначения сетей любых предприятий и организаций.

Главными особенностями сетей кампусов являются следующие. Сети этого типа объединяют множество сетей различных отделов одного предприятия в пределах отдельного здания или в пределах одной территории, покрывающей площадь в несколько квадратных километров. При этом глобальные соединения в сетях кампусов не используются. Службы такой сети включают взаимодействие между сетями отделов, доступ к общим базам данных предприятия, доступ к общим факс-серверам, высокоскоростным модемам и высокоскоростным принтерам. В результате сотрудники каждого отдела предприятия получают доступ к некоторьм файлам и ресурсам сетей других отделов. Важной службой, предоставляемой сетями кампусов, стал доступ к корпоративным базам данных независимо от того, на каких типах компьютеров они располагаются.

Именно на уровне сети кампуса возникают проблемы интеграции неоднородного аппаратного и программного обеспечения. Типы компьютеров, сетевых операционных систем, сетевого аппаратного обеспечения могут отличаться в каждом отделе. Отсюда вытекают сложности управления сетями

кампусов. Администраторы должны быть в этом случае более квалифицированными, а средства оперативного управления сетью - более совершенными.

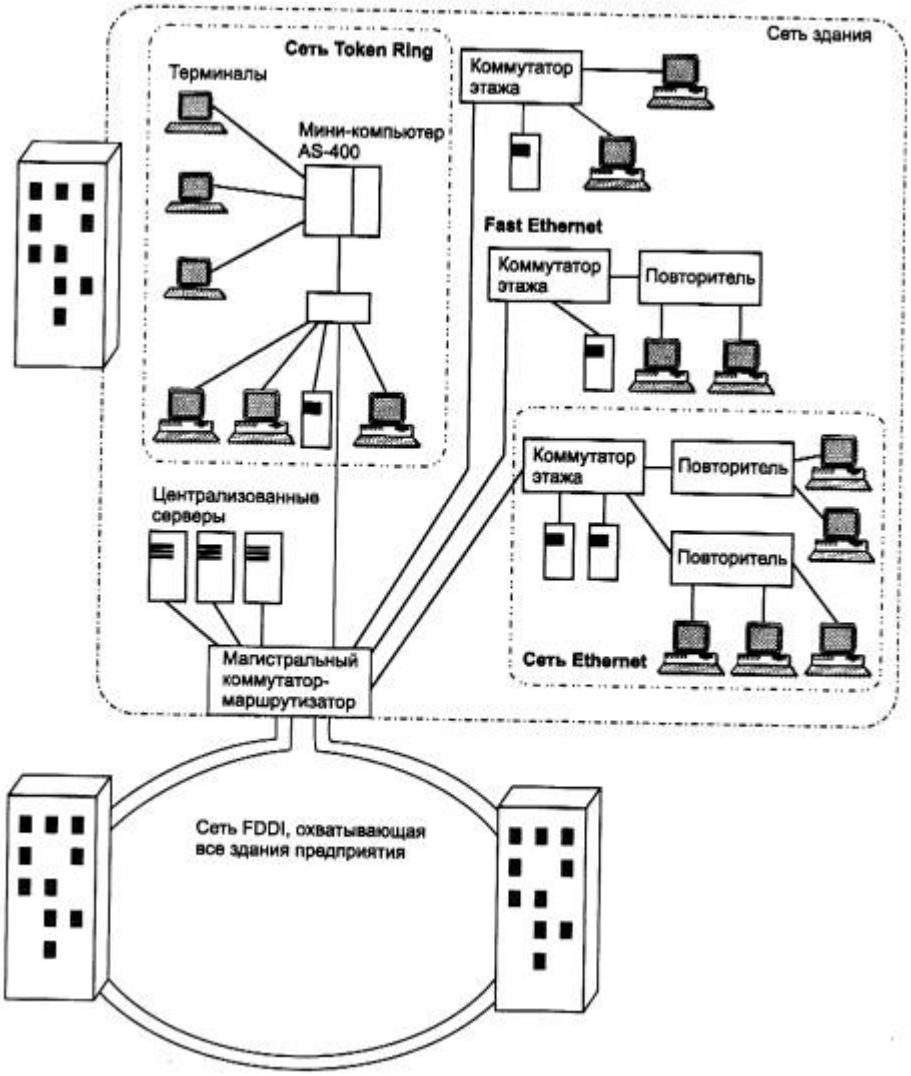


Рисунок 10 – Пример сети кампуса

Корпоративные сети называют также сетями масштаба предприятия. Сети масштаба предприятия (корпоративные сети) объединяют большое количество компьютеров на всех территориях отдельного предприятия. Они могут быть сложно связаны и покрывать город, регион или даже континент. Число пользователей и компьютеров может измеряться тысячами, а число серверов - сотнями, расстояния между сетями отдельных территорий могут оказаться такими, что становится необходимым использование глобальных связей. Для соединения удаленных локальных сетей и отдельных компьютеров в корпоративной сети применяются разнообразные телекоммуникационные средства, в том числе телефонные каналы, радиоканалы, спутниковую связь. Корпоративную сеть можно представить в виде «островков локальных сетей», плавающих в телекоммуникационной среде.

В корпоративной сети обязательно будут использоваться различные типы компьютеров - от мэйнфреймов до персоналок, несколько типов операционных

систем и множество различных приложений. Неоднородные части корпоративной сети должны работать как единое целое, предоставляя пользователям по возможности прозрачный доступ ко всем необходимым ресурсам.

При объединении отдельных сетей крупного предприятия, имеющего филиалы в разных городах и даже странах, в единую сеть многие количественные характеристики объединенной сети превосходят некоторый критический порог, за которым начинается новое качество.

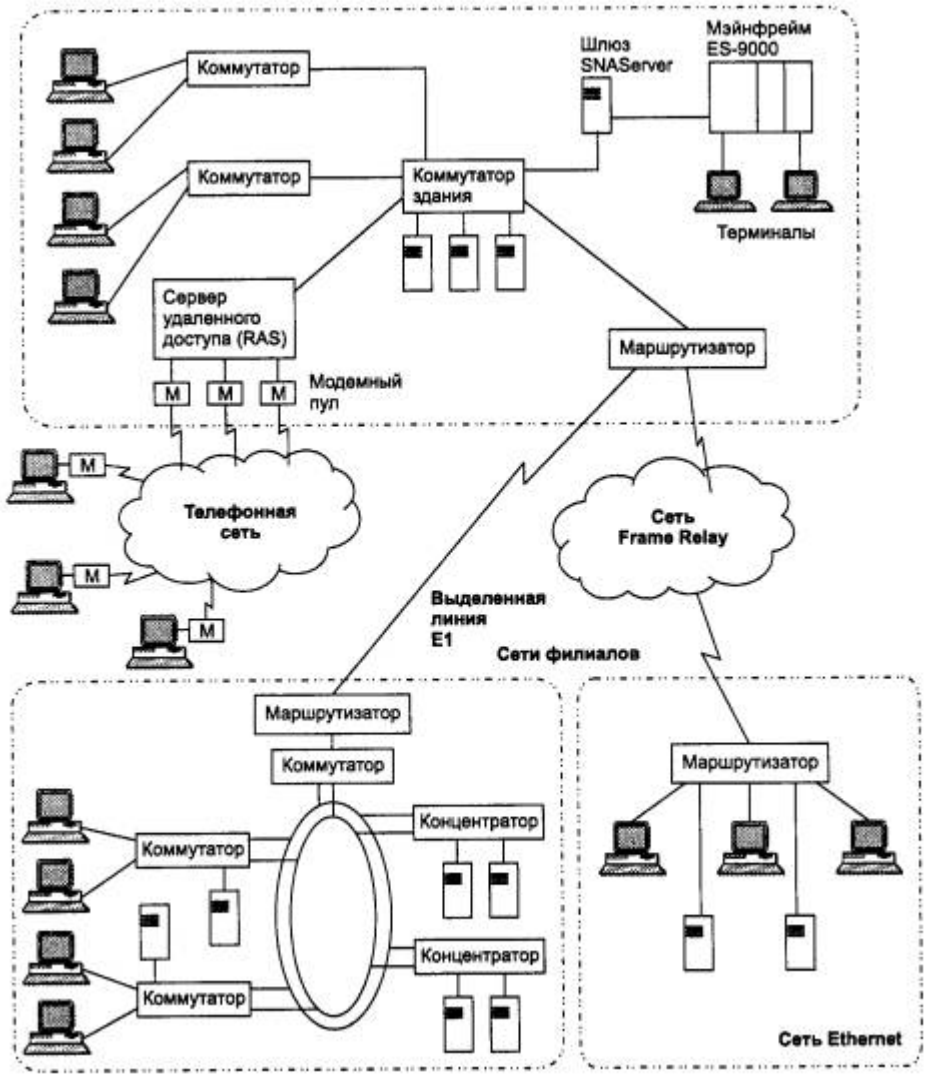


Рисунок 11 – Пример корпоративной сети

Наиболее простой способ ее решения - помещение учетных данных каждого пользователя в локальную базу учетных данных каждого компьютера, к ресурсам которого пользователь должен иметь доступ. При попытке доступа эти данные извлекаются из локальной учетной базы и на их основе доступ предоставляется или не предоставляется. Для небольшой сети, состоящей из 5-10 компьютеров и примерно такого же количества пользователей, такой способ работает очень хорошо. Но если в сети насчитывается несколько тысяч пользователей, каждому из которых нужен доступ к нескольким десяткам

серверов, то, очевидно, это решение становится крайне неэффективным. Администратор должен повторить несколько десятков раз операцию занесения учетных данных пользователя. Сам пользователь также вынужден повторять процедуру логического входа каждый раз, когда ему нужен доступ к ресурсам нового сервера. Хорошее решение этой проблемы для крупной сети - использование централизованной справочной службы, в базе данной которой хранятся учетные записи всех пользователей сети. Администратор один раз выполняет операцию занесения данных пользователя в эту базу, а пользователь один раз выполняет процедуру логического входа, причем не в отдельный сервер, а в сеть целиком.

При переходе от более простого типа сетей к более сложному - от сетей отдела к корпоративной сети - сеть должна быть все более надежной и отказоустойчивой, при этом требования к ее производительности также существенно возрастают. По мере увеличения масштабов сети увеличиваются и ее функциональные возможности. По сети циркулирует все возрастающее количество данных, и сеть должна обеспечивать их безопасность и защищенность наряду с доступностью. Соединения, обеспечивающие взаимодействие, должны быть более прозрачными. При каждом переходе на следующий уровень сложности компьютерное оборудование сети становится все более разнообразным, а географические расстояния увеличиваются, делая достижение целей более сложным; более проблемным и дорогостоящим становится управление такими соединениями.

1.4 Сетевые топологии и методы доступа к среде передачи данных

Топология сети характеризует взаимосвязи и пространственное расположение друг относительно друга компонентов сети - сетевых компьютеров (хостов), рабочих станций, кабелей и других активных и пассивных устройств. Топология влияет на:

- состав и характеристики оборудования сети;
- возможности расширения сети;
- способ управления сетью.

Все сети строятся на основе трех базовых топологий:

- шина (bus);
- звезда (star);
- кольцо (ring).

Метод доступа к среде передачи данных определяет, каким образом разделяемый ресурс - сетевой кабель - предоставляется узлам сети для осуществления актов передачи данных. Основные методы доступа к среде передачи данных:

- состязательный метод (множественный доступ с контролем несущей и обнаружением коллизий - CSMA/CD);
- с передачей маркера;
- по приоритету запроса.

1.4.1 Шинная топология

При помощи кабеля каждая рабочая станция соединяется с другими рабочими станциями и с файловым сервером. Кабель проходит от узла к узлу, последовательно соединяя все рабочие станции и все файловые серверы. На каждом конце кабеля подключается согласующая нагрузка (терминатор) для исключения эхоотражений (рисунок 12).

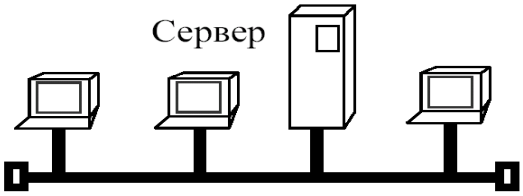


Рисунок 12 – Шинная топология

Шинная топология использует состязательный метод доступа. Это означает, что информацию принимает только тот компьютер, адрес которого соответствует адресу получателя, зашифрованному в передаваемых сигналах. Остальные компьютеры отбрасывают сообщение. Перед передачей данных компьютер должен ожидать освобождения шины. В каждый момент времени отправлять сообщение может только один компьютер, поэтому число подключенных к сети машин значительно влияет на ее быстродействие.

Преимущества шинной топологии:

- надежно работает в небольших сетях, проста в использовании;
- требует меньше кабеля для соединения компьютеров и потому дешевле, чем другие схемы соединении;
- легко расширяется за счет состыковки кабельных сегментов с помощью цилиндрического соединителя BNC и использования повторителей.

Недостатки шинной топологии:

- интенсивный сетевой трафик снижает производительность сети. При большом числе компьютеров в сети станции часто прерывают друг друга, и немалая часть полосы пропускания теряется понапрасну. При добавлении компьютеров к сети резко падает производительность;
- цилиндрические соединители ослабляют электрический сигнал, и большое их число вызывает нарушения в передаче информации по шине;
- разрыв кабеля или неправильное функционирование одной из станций может привести к нарушению работоспособности всей сети. Сеть трудно диагностировать.

1.4.2 Звездообразная топология

Каждый компьютер в сети с топологией типа “звезда” (“star”) взаимодействует с центральным концентратором (hub – устройство для повторения сетевых сигналов) (рисунок 13).

Рисунок 13 – Топология “звезда”

Hub - устройство множественного доступа, выполняющее роль центральной точки соединения в топологии "физическая звезда". Наряду с традиционным названием "концентратор" в литературе встречается также термин "хаб".

В звездообразной сети используется состязательный метод доступа к среде - концентратор (хаб) передает сообщение всем компьютерам. В звездообразной сети с коммутацией коммутатор передает сообщение только компьютеру-адресату /2/.

Активный концентратор регенерирует электрический сигнал и посылает его всем подключенным компьютерам. Такой тип концентратора часто называют многопортовым повторителем (multiport repeater). Для работы активных концентраторов и коммутаторов требуется питание от сети. Пассивные концентраторы, например, коммутационная кабельная панель или коммутационный блок, действуют как точка соединения, не усиливая и не регенерируя сигнал. Электропитания пассивные концентраторы не требуют.

Гибридный концентратор позволяет использовать в одной звездообразной сети разные типы кабелей. Расширить звездообразную сеть можно путем подключения вместо одного из компьютеров еще одного концентратора и подсоединения к нему дополнительных станций, в результате чего получается гибридно-звездообразная сеть (рисунок 14).

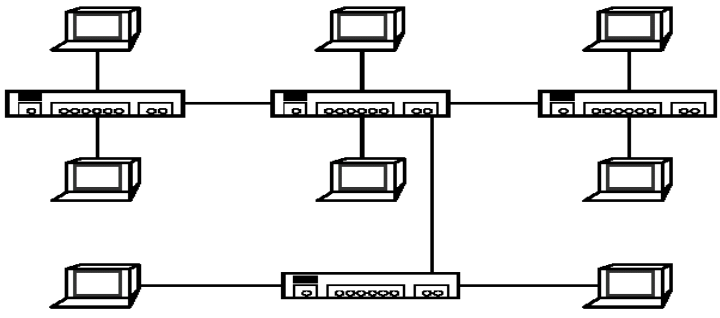


Рисунок 14 – Гибридно-звездообразная топология

Преимущества топологии "звезда "(Ethernet 10BaseT, 100BaseT).

Центральный концентратор звездообразной сети удобно использовать для диагностики. *Интеллектуальные концентраторы* (устройства с микро-

процессорами, добавленными для повторения сетевых сигналов) обеспечивают также измерение параметров (мониторинг) и управление сетью. Отказ одного компьютера не обязательно приводит к останову всей сети. Концентратор способен выявлять отказы и изолировать такую машину или сетевой кабель, что позволяет остальной сети продолжать работу. В одной сети допускается применение нескольких типов кабелей (если их позволяет использовать концентратор).

Недостатки сети со звездообразной топологией:

- при отказе центрального концентратора вся сеть становится неработоспособной;
- все компьютеры должны соединяться с центральной точкой, это увеличивает расход кабеля, следовательно, такие сети обходятся дороже, чем сети с иной топологией.

1.4.3 Кольцевая топология

На рисунке 15 показан пример топологии ЛВС, в которой каждая рабочая станция соединена с двумя другими рабочими станциями. Такая топология называется кольцом (ring).

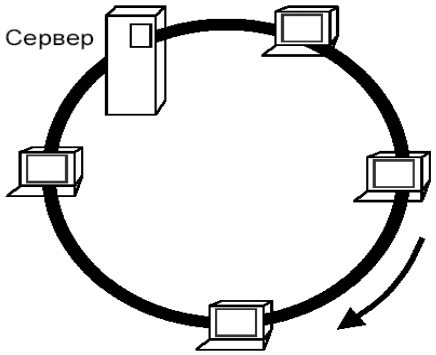


Рисунок 15 – Кольцевая топология

Кольцевая топология применяется преимущественно в США для сетей, требующих выделения определенной части полосы пропускания для критичных по времени средств (например, для передачи видео и аудио), в высокопроизводительных сетях, а также при большом числе обращающихся к сети клиентов (что требует ее высокой пропускной способности). В сети с кольцевой топологией каждый компьютер соединяется со следующим компьютером, ретранслирующим ту информацию, которую он получает от первой машины. Благодаря такой ретрансляции сеть является активной, и в ней не возникают проблемы потери сигнала, как в сетях с шинной топологией. Кроме того, поскольку «конца» в кольцевой сети нет, никаких оконечных нагрузок не нужно.

Некоторые сети с кольцевой топологией используют метод доступа к среде на основе маркера (метод эстафетной передачи). Специальное короткое сообщение-маркер циркулирует по кольцу пока компьютер не пожелает передать информацию другому узлу. Он модифицирует маркер, добавляет элек-

тронный адрес и данные, а затем отправляет его по кольцу. Каждый из компьютеров последовательно получает данный маркер с добавленной информацией и передает его соседней машине, пока электронный адрес не совпадет с адресом компьютера-получателя, или маркер не вернется к отправителю. Получивший сообщение компьютер возвращает отправителю ответ, подтверждающий, что послание принято. Тогда отправитель создает еще один маркер и отправляет его в сеть, что позволяет другой станции перехватить маркер и начать передачу. Маркер циркулирует по кольцу, пока какая-либо из станций не будет готова к передаче и не захватит его.

Все эти события происходят очень часто: маркер может пройти кольцо с диаметром в 200 м примерно 10000 раз в секунду. В некоторых еще более быстрых сетях циркулирует сразу несколько маркеров. В других сетевых средах применяются два кольца с циркуляцией маркеров в противоположных направлениях. Такая структура способствует восстановлению сети в случае возникновения отказов.

Преимущества сети с кольцевой топологией:

- поскольку всем компьютерам предоставляется равный доступ к маркеру, никто из них не сможет монополизировать сеть;
- справедливое совместное использование сети обеспечивает постепенное снижение ее производительности в случае увеличения числа пользователей и перегрузки (лучше, если сеть будет продолжать функционировать, хотя и медленно, чем сразу откажет при превышении пропускной способности).

Недостатки сети с кольцевой топологией:

- отказ одного компьютера в сети может повлиять на работоспособность всей сети;
- кольцевую сеть трудно диагностировать;
- добавление или удаление компьютера вынуждает разрывать сеть.

1.4.4 Смешанные топологии

На основе трех базовых топологий можно создавать так называемые гибридные или смешанные топологии. К этим топологиям относятся:

- шинно-звездообразная;
- звездообразно-кольцевая.

Шинно-звездообразная топология комбинирует сети типа «звезда» и «шина», связывая несколько концентраторов шинными магистралями (рисунок 16).

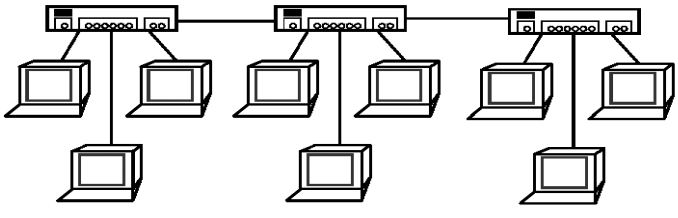


Рисунок 16 – Шинно-звездообразная топология

Если один из компьютеров отказывает, концентратор может выявить отказавший узел и изолировать неисправную машину. При отказе концентратора соединенные с ним компьютеры не смогут взаимодействовать с сетью, а шина разомкнется на два не связанных друг с другом сегмента /2/.

В звездообразно-кольцевой топологии (которую называют также кольцом с соединением типа «звезда») сетевые кабели прокладываются аналогично звездообразной сети, но в центральном концентраторе реализуется кольцо (рисунок 17).

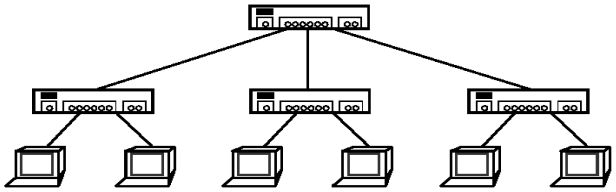


Рисунок 17 – Звездообразно-кольцевая топология

С внутренним концентратором можно соединить внешние, тем самым, расширив петлю внутреннего кольца. Большие объединенные ВС используют топологию самого общего вида - *ячеистую.* Узлами ячеистой топологии могут быть самые разнообразные сетевые устройства: повторители, мосты, концентраторы, маршрутизаторы, шлюзы.

1.5 Основные типы кабельных сред передачи данных

На сегодня, большая часть компьютерных сетей используют для соединения провода и кабели. Они выступают в качестве среды передачи сигналов между компьютерами. Наиболее распространены: коаксиальный кабель, витая пара, оптоволоконный кабель (таблица 2).

Таблица 2 - Сетевые кабели

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

Однако постепенно в нашу жизнь входит беспроводная среда передачи данных. Для беспроводной передачи данных используют: инфракрасное и

лазерное излучение, радиопередачу и телефонию. Эти способы передачи данных в компьютерных сетях, как локальных, так и глобальных, привлекательны тем, что: гарантируют определенный уровень мобильности; позволяют снять ограничение на длину сети, а использование радиоволн и спутниковой связи делают доступ к сети фактически неограниченным.

В таблице 2 приведены основные типы кабелей, используемых в ЛВС. Для высокопроизводительного обмена, но на ограниченном расстоянии, развивалось несколько направлений реализации локальных сетей - Ethernet, ARCnet, TokenRing, адаптеры которых широко используются в персональных компьютерах (ПК).

1.5.1 Коаксиальный кабель

До недавнего времени самой распространенной средой передачи данных был коаксиальный кабель: относительно недорогой, легкий и гибкий, безопасный и простой в установке. На рисунке 18 приведена конструкция коаксиального кабеля /6/.

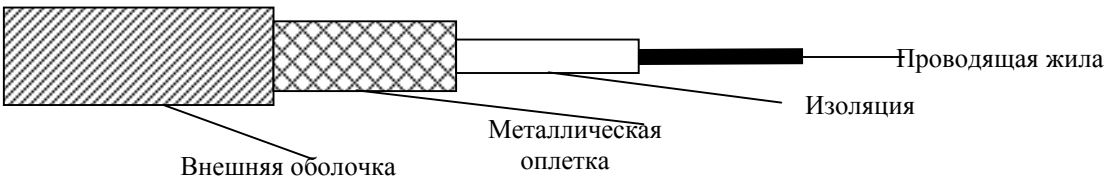


Рисунок 18 - Конструкция коаксиального кабеля

Электрические сигналы, кодирующие данные, передаются по жиле. Она изоляцией отделяется от металлической оплетки, которая играет роль заземления и защищает передаваемые по жиле сигналы от:

- внешних электромагнитных шумов (атмосферных, промышленных);
- перекрестных помех – электрических наводок, вызванных сигналами в соседних проводах.

Используют толстый и тонкий коаксиальный кабель. Их характеристики представлены в таблице 3.

Таблица 3 - Характеристики коаксиального кабеля

Тип	Диаметр	Эффективная длина сегмента	Скорость передачи	Обозначение по стандарту IEEE 802.3
толстый	1 см	500 м	10 Мбит/с	10 base 5
тонкий	0,5 см	185 м	10 Мбит/с	10 base 2

В обозначении кабелей по стандарту IEEE 802.3 первые две цифры – скорость передачи в Мбит/с, base обозначает, что кабель используется в сетях с

узкополосной передачей (baseband network), последняя цифра – эффективная длина сегмента в сотнях метров, при которой уровень затухания сигнала остается в допустимых пределах. Тонкий подключается к сетевым платам непосредственно через *Т-коннектор* (рисунок 19), толстый – через специальное устройство - *трансивер* (рисунок 20).

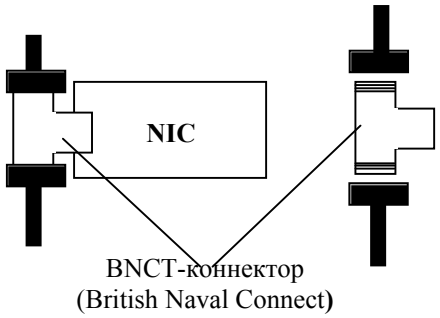


Рисунок 19 - Подключение тонкого коаксиального кабеля

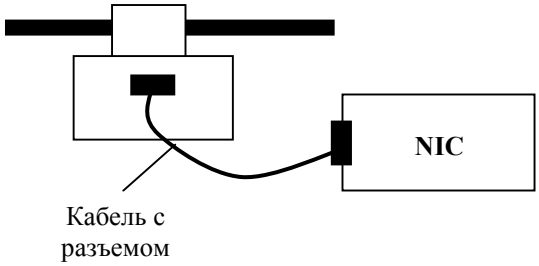


Рисунок 20 – Подключение толстого коаксиального кабеля

Различают *обычные* и *пленумные* коаксиальные кабели. Последние обладают повышенными механическими и противопожарными характеристиками и допускают прокладку под полом, между фальшпотолком и перекрытием. При выборе для ЛВС данного типа кабеля следует принимать во внимание, что:

1) это среда для передачи речи, видео и двоичных данных;

2) позволяет передавать данные на большие расстояния;

3) это хорошо знакомая технология, предлагающая достаточный уровень защиты данных.

1.5.2 Витая пара

Если для передачи электрических сигналов воспользоваться обычной парой параллельных проводов для передачи знакопеременного списка большой частоты, то возникающие вокруг одного из них магнитные потоки будут вызывать помехи в другом (рисунок 21). Для исключения этого явления провода перекручивают между собой (рисунок 22).

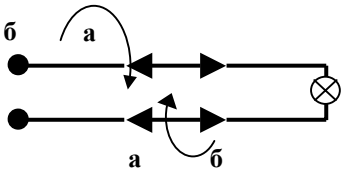


Рисунок 21 - Пара параллельных проводов

Рисунок 22 - Витая пара

Самая простая *витая пара (twisted pair)* – это два перевитых друг вокруг друга изолированных провода. Существует два вида такого кабеля:

– неэкранированная витая пара (UTP);
– экранированная витая пара (STP).

Часто несколько витых пар помещают в одну защитную оболочку (типа телефонного кабеля). Наиболее распространена в ЛВС неэкранированная витая пара стандарта 10 baseT с эффективной длиной сегмента – 100 м. Определено 5 категорий на основе UTP (таблица 4).

Таблица 4 - Категории кабельных соединений на неэкранированной витой паре

Категория	Скорость передачи (Мбит/с)	Количество пар
1	Телефонный кабель только для передачи речи	1 пара
2	До 4	4 пары
3	До 10	4пары с 9-ю витками на 1 м
4	До 16	4 пары
5	До 100	4 медных пары

Одной из проблем всех этих кабелей являются перекрестные помехи, т.е. наводки со стороны соседних линий, что может приводить к искажению передаваемых данных. Для уменьшения их влияния используют экран. В кабелях на основе экранированных витых пар каждая пара обматывается фольгой, а сам кабель заключается в медную оплетку, что позволяет передавать данные с более высокой скоростью и на большие расстояния.

1.5.3 Оптоволоконный кабель

В оптоволоконном кабеле цифровые данные распространяются по оптическим волокнам в виде модулированных световых импульсов, а не электрических сигналов. Следовательно, его нельзя вскрыть и перехватить данные.

Передача по оптоволоконному кабелю не подвержена электрическим помехам и ведется на чрезвычайно высокой скорости (до 100 Мбит/с, а теоретически возможно до 200 Мбит/с).

Основа кабеля – оптическое волокно – тонкий стеклянный цилиндр (жила), покрытая слоем стекла, называемого оболочкой и имеющей отличный от жилы коэффициент преломления (рисунок 23).

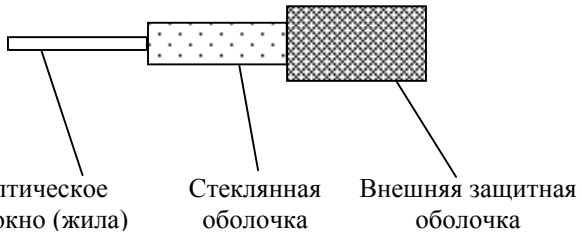


Рисунок 23 – Структура оптоволокна

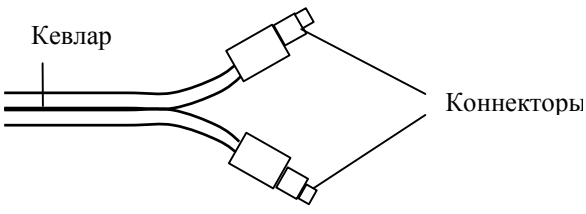


Рисунок 24 - Оптоволоконный кабель

Каждое стеклянное оптоволокно передает сигналы только в одном направлении, поэтому кабель состоит из двух волокон с отдельными коннекторами (рисунок 24). Жесткость обеспечивает покрытие из пластика, а прочность – волокна кевлара. Оптоволоконный кабель рекомендуется использовать при передаче данных на большие расстояния с высокой скоростью по надежной среде передачи.

Не рекомендуется использовать:

- при ограниченности денежных средств;
- при отсутствии навыков установки и корректного подключения оптоволоконных сетевых устройств.

1.6 Контрольные вопросы

1) Основное назначение вычислительных сетей
2) Принцип работы и недостатки технологии “клиент-сервер”.
3) Особенности различных топологии.
4) Типы соединительных кабелей.
5) Классификация сетей.

1.7 Тесты

1) Из чего состоит самая простая сеть?

а) из нескольких персональных компьютеров, соединенных между собой сетевым кабелем;

б) из 2 персональных компьютеров, соединенных между собой 0–модемным кабелем;

в) из нескольких ЭВМ, один из которых обязательно наделяется правами сервера.

2) Что означает параллельная передача данных?

а) данные передаются одновременно по нескольким проводам;

б) данные передаются поочередно бит за битом.

3) Принцип архитектуры “клиент-сервер”:

а) существует выделенный сервер, предоставляющий всевозможные сервисы, и множество клиентских ПК, использующих их в своих целях;

б) каждый ПК является как сервером, так и клиентом;

в) ни один из ПК не обладает полномочиями сервера.

4) Одноранговые сети – это:

а) сети с одним выделенным сервером;

б) сети с одним и более выделенными серверами;

в) сети, где все компьютеры равноправны.

5) Технологии передачи данных, используемые в сетях:

а) точка-точка;
б) передача;
в) вещание (от одного ко многим).

6) Сети отделов – это:
а) локальные сети, имеющие выход в глобальную сеть Internet;
б) сети, которые используются сравнительно небольшой группой сотрудников, работающих в одном отделе предприятия.
в) локальные сети, не имеющие выход в глобальную сеть InterNet и функционирующие без выделенного сервера.

7) Сети кампусов – это:
а) сети, объединяющие множество сетей различных отделов одного предприятия в пределах отдельного здания или в пределах одной территории:
б) подсети сетей отделов;
в) локальные сети, не имеющие выход в глобальную сеть Internet и функционирующие без выделенного сервера.

8) Сетевые кабели, обладающие наибольшей скоростью и качеством передачи данных:
а) витая пара;
б) опто-волокно;
в) коаксиальный кабель.

9) Какая из топологий использует метод доступа к среде на основе маркера:
а) звезда;
б) шина;
в) кольцо.

10) Какая из тополгоий не относится к смешанным?
а) шинно-звездообразная;
б) звездообразно-кольцевая;
в) шинно-кольцевая.

11) Другое название концентратора:
а) Hub;
б) Switch;
в) Router.

12) Эффективная длина сетевого кабеля витая пара?
а) 50 м;
б) 100 м;
в) 150 м;
г) 500 м.

13) Для чего скручивают провода витой пары:
а) чтобы компактнее разместить их в защитной оболочке;
б) для уменьшения помех, вызванных магнитными потоками;
в) для четкого разделения каждой пары проводов.

14) Стеклянное оптоволокно передает сигналы:
а) в одном направлении;
б) в двух направлениях.

15) Через какое устройство подключается тонкий коаксиальный кабель?
а) трансивер;
б) Т-коннектор;
в) повторитель;
г) хаб.

2 Взаимодействие открытых систем

2.1 Эталонная модель OSI

Обмен информацией между компьютерами, объединенными в сеть, очень сложная задача. Это связано с тем, что существует много производителей аппаратных и программных средств вычислительных систем. Единственный выход - унифицировать средства сопряжения систем, а именно использовать открытые системы. Открытая система взаимодействует с другими системами в соответствии с принятыми стандартами.

В 1984г. Международная Организация по Стандартизации (ISO) выпустила стандарт - *семиуровневую эталонную модель взаимодействия открытых систем* (Seven-layer Open System Interconnection Reference Model - OSI, Эталонная модель взаимодействия открытых систем), чтобы помочь поставщикам создавать совместимые сетевые аппаратные и программные средства. Модель OSI представляет собой универсальный стандарт на взаимодействие двух систем (компьютеров) через вычислительную сеть /7/.

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

Каждый уровень поддерживает интерфейсы с выше- и нижележащими уровнями.

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

Уровень приложения (Application) - интерфейс с прикладными процессами.

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

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

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

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

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

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

Принципы выделения этих уровней таковы: каждый уровень отражает надлежащий уровень абстракции. Каждый уровень имеет строго определенную

функцию. Эта функция выбиралась, прежде всего, так, чтобы можно было определить международный стандарт. Границы уровней выбирались так, чтобы минимизировать поток информации через интерфейсы.

Два самых низших уровня - физический и канальный - реализуются аппаратными и программными средствами, остальные пять более высоких уровней реализуются, как правило, программными средствами (рисунок 25).

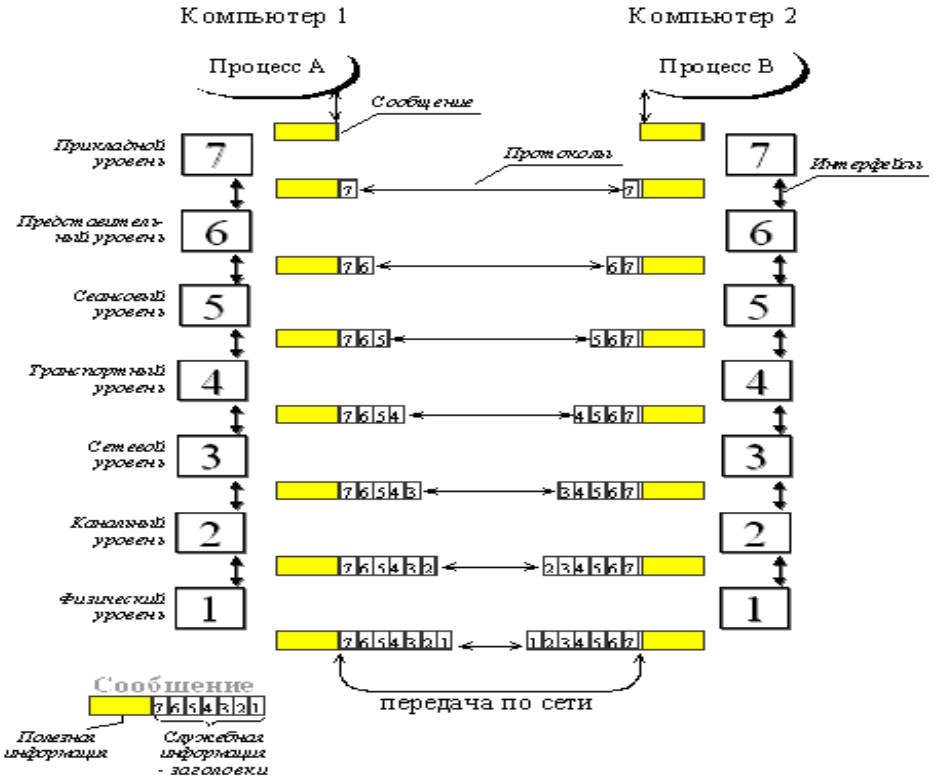


Рисунок 25 - Модель взаимодействия открытых систем ISO/OSI

При продвижении пакета данных по уровням сверху вниз каждый новый уровень добавляет к пакету свою служебную информацию в виде заголовка и, возможно, трейлера (информации, помещаемой в конец сообщения). Эта операция называется инкапсуляцией данных верхнего уровня в пакете нижнего уровня. Служебная информация предназначается для объекта того же уровня на удаленном компьютере, ее формат и интерпретация определяются протоколом данного уровня.

Когда сообщение по сети поступает на другую машину, оно последовательно перемещается вверх с уровня на уровень. Каждый уровень анализирует, обрабатывает и удаляет заголовок своего уровня, выполняет соответствующие данному уровню функции и передает сообщение вышележащему уровню. Тот в свою очередь рассматривает эти данные как пакет со своей служебной информацией и данными для более верхнего уровня, и процедура повторяется, пока пользовательские данные, очищенные от всей служебной информации, не достигнут прикладного процесса /7/.

Кроме термина "сообщение" (message) существуют и другие названия, используемые сетевыми специалистами для обозначения единицы обмена

данными. В стандартах ISO для протоколов любого уровня используется такой термин как "протокольный блок данных" - Protocol Data Unit (PDU). Кроме этого, часто используются названия кадр (frame), пакет (packet), дейтаграмма (datagram).

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

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

Этот уровень отвечает за передачу последовательности битов через канал связи. Основной проблемой является, как гарантировать, что если на одном конце послали 1, то на другом получили 1, а не 0. На этом уровне решают такие вопросы, каким напряжением надо представлять 1, а каким - 0; сколько микросекунд тратиться на передачу одного бита; следует ли поддерживать передачу данных в обоих направлениях одновременно; как устанавливается начальное соединение и как оно разрывается; каково количество контактов на сетевом разъеме, для чего используется каждый контакт. Здесь в основном вопросы механики, электрики.

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

Примером протокола физического уровня может служить спецификация 10Base-T технологии Ethernet.

Канальный уровень. Основной задачей уровня канала данных - превратить несовершенную среду передачи в надежный канал, свободный от ошибок передачи. Другой задачей канального уровня является реализация механизмов обнаружения и коррекции ошибок. Эта задача решается разбиением данных отправителя на фреймы (обычно от нескольких сотен до нескольких тысяч байтов), передачей фреймов последовательно и обработкой фреймов уведомления, поступающих от получателя. Поскольку физический уровень не распознает структуры в передаваемых данных, то это целиком и полностью задача канала данных определить границы фрейма. Эта задача решается введением специальной последовательности битов, которая добавляется в начало и в конец фрейма и всегда интерпретируется как границы фрейма.

Помехи на линии могут разрушить фрейм. В этом случае он должен быть передан повторно. Он будет повторен также и в том случае если фрейм уведомление будет потерян. И это уже заботы уровня как бороться с

дубликатами одного и того же фрейма, потерями или искажениями фреймов. Уровень канала данных может поддерживать сервис разных классов для сетевого уровня, разного качества и стоимости.

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.

На физическом уровне просто пересылаются биты. При этом не учитывается, что в некоторых сетях, в которых линии связи используются (разделяются) попеременно несколькими парами взаимодействующих компьютеров, физическая среда передачи может быть занята. Поэтому одной из задач канального уровня является проверка доступности среды передачи. В локальных сетях протоколы канального уровня используются компьютерами, мостами, коммутаторами и маршрутизаторами. В компьютерах функции канального уровня реализуются совместными усилиями сетевых адаптеров и их драйверов.

Примерами протоколов "точка - точка" (как часто называют такие протоколы) могут служить широко распространенные протоколы PPP и LAP-B, Ethernet, Token Ring, FDDI.

Сетевой уровень. Этот уровень обеспечивает возможность соединения и выбор маршрута между двумя конечными системами, подключенными к разным подсетям. Маршруты могут быть определены заранее и прописаны в статической таблице, которая не изменяется. Они могут определяться в момент установления соединения. Наконец, они могут строиться динамически в зависимости от загрузки сети.

Если в подсети циркулирует слишком много пакетов, то они могут использовать одни и те же маршруты, что будет приводить к заторам. Эта проблема так же решается на сетевом уровне.

Поскольку за использование подсети, как правило, предполагается оплата, то на этом уровне также присутствуют функции учета: как много байт, символов послал или получил абонент сети. Если абоненты расположены в разных странах, где разные тарифы, то надо должным образом скорректировать цену услуги.

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

Внутри сети доставка данных регулируется канальным уровнем, а вот доставкой данных между сетями занимается сетевой уровень.

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.

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.

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

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

Основная функция транспортного уровня это: принять данные с уровня сессии, разделить, если надо, на более мелкие единицы, передать на сетевой уровень и позаботиться, чтобы все они дошли в целостности до адресата. Все это должно быть сделано эффективно и так, чтобы скрыть от вышележащего уровня непринципиальные изменения на нижних.

Транспортный уровень определяет, какой тип сервиса предоставить вышележащим уровням и пользователям сети. Наиболее часто используемым сервисом является канал точка-точка без ошибок, обеспечивающий доставку сообщений или байтов в той последовательности, в какой они были отправлены. Другой вид сервиса - доставка отдельных сообщений без гарантии сохранения их последовательности, рассылка одного сообщения многим в режиме вещания. Тип сервиса определяется при установлении транспортного соединения.

Транспортный уровень также отвечает за установление и разрыв транспортного соединения в сети. Это предполагает наличие механизма именования, т.е. процесс на одной машине должен уметь указать с кем в сети ему надо обменяться информацией. Транспортный уровень также должен предотвращать «захлебывание» получателя в случае очень «быстро говорящего» отправителя. Механизм для этого называется управление потоком. Он есть и на других уровнях. Однако управление потоком между хостами отличен от управления потоком между маршрутизаторами, хотя у них есть общие принципы.

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. Уровень сессии позволяет пользователям на разных машинах (напомним, что пользователем может быть программа) устанавливать

сессии. Сессия позволяет передавать данные, как это может делать транспортный уровень, но кроме этого этот уровень имеет более сложный сервис, полезный в некоторых приложениях. Например, вход в удаленную систему, передать файл между двумя приложениями.

Одним из видов услуг на этом уровне - управление диалогом. Потоки данных могут быть разрешены в обоих направлениях одновременно, либо поочередно в одном направлении. Сервис на уровне сессии будет управлять направлением передачи.

Другим видом сервиса - управление маркером. Для некоторых протоколов недопустимо выполнение одной и той же операции на обоих концах соединения одновременно. Для этого уровень сессии выделяет активной стороне маркер. Операцию может выполнять тот, кто владеет маркером.

Другой услугой уровня сессии является синхронизация. Пусть нам надо передать файл такой, что его пересылка займет два часа, между машинами, время наработки на отказ, у которых один час. Ясно что «в лоб» такой файл средствами транспортного уровня не решить. Уровень сессии позволяет расставлять контрольные точки. В случае отказа одной из машин передача возобновиться с последней контрольной точки.

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.

Представительный уровень. Представительный уровень (уровень представления) определяет синтаксис, форматы и структуры представления передаваемых данных (но не затрагивает семантику, значение данных). Для того чтобы информация, посылаемая из прикладного уровня одной системы, была читаемой на прикладном уровне другой системы, представительный уровень осуществляет трансляцию между известными форматами представления информации за счет использования общего формата. Этот уровень имеет дело с информацией, а не с потоком битов.

Типичным примером услуги на этом уровне - унифицированная кодировка данных. Дело в том, что на разных машинах используются разные способы кодировки ASCII, Unicode и т.п. для символов, разные способы представления целых - в прямом, обратном или дополнительном коде, нумирация бит в байте слева направо или наоборот и т.п. Пользователи, как правило, используют структуры данных, а не случайный набор байт. Для того, чтобы машины с разной кодировкой и представлением данных могли взаимодействовать, передаваемые структуры данных определяются специальным абстрактным способом, не зависящим от кодировки, используемой при передачи. Уровень представления работает со структурами данных в абстрактной форме, преобразует это представление во внутреннее для

конкретной машины и из внутреннего, машинного представления в стандартное представление для передачи по сети.

На этом уровне может выполняться шифрование и дешифрование данных, благодаря которому секретность обмена данными обеспечивается сразу для всех прикладных сервисов.

Прикладной уровень. В отличие от других уровней прикладной уровень - самый близкий к пользователю уровень OSI - не предоставляет услуги другим уровням OSI, однако он обеспечивает прикладные процессы, лежащие за пределами масштаба модели OSI.

Прикладной уровень обеспечивает непосредственную поддержку прикладных процессов и программ конечного пользователя (программ обработки крупномасштабных таблиц, текстовых процессоров, программ банковских терминалов и многое другое) и управление взаимодействием этих программ с сетью передачи данных.

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.

Модель OSI представляет хотя и очень важную, но только одну из многих моделей коммуникаций. Эти модели и связанные с ними стеки протоколов могут отличаться количеством уровней, их функциями, форматами сообщений, сервисами, предоставляемыми на верхних уровнях и прочими параметрами.

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

2.2 Характеристика стеков коммуникационных протоколов

Главная цель, которая преследуется при соединении компьютеров в сеть - это возможность использования ресурсов каждого компьютера всеми пользователями сети. Для того, чтобы реализовать эту возможность, компьютеры, подсоединенные к сети, должны иметь необходимые для этого средства взаимодействия с другими компьютерами сети. Задача разделения сетевых ресурсов является сложной.

Обычным подходом при решении сложной проблемы является ее декомпозиция на несколько частных проблем - подзадач. Для решения каждой подзадачи назначается некоторый модуль. При этом четко определяются функции каждого модуля и правила их взаимодействия /8/.

Частным случаем декомпозиции задачи является многоуровневое представление, при котором все множество модулей, решающих подзадачи, разбивается на иерархически упорядоченные группы - уровни. Для каждого уровня определяется набор функций-запросов, с которыми к модулям данного уровня могут обращаться модули выше лежащего уровня для решения своих

задач. Такой формально определенный набор функций, выполняемых данным уровнем для выше лежащего уровня, а также форматы сообщений, которыми обмениваются два соседних уровня в ходе своего взаимодействия, называется *интерфейсом*.

Интерфейс определяет совокупный сервис, предоставляемый данным уровнем выше лежащему уровню.

При организации взаимодействия компьютеров в сети каждый уровень ведет "переговоры" с соответствующим уровнем другого компьютера. При передаче сообщений оба участника сетевого обмена должны принять множество соглашений. Соглашения должны быть приняты для всех уровней, начиная от самого низкого уровня передачи битов, до самого высокого уровня, детализирующего, как информация должна быть интерпретирована.

Правила взаимодействия двух машин могут быть описаны в виде набора процедур для каждого из уровней. Такие формализованные правила, определяющие последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на одном уровне, но в разных узлах, называются *протоколами*.

Протоколы реализуются не только программно-аппаратными средствами компьютеров, но и коммуникационными устройствами. В общем случае связь компьютеров в сети осуществляется не напрямую - "компьютер-компьютер", а через различные коммуникационные устройства такие, например, как концентраторы, коммутаторы или маршрутизаторы. В зависимости от типа устройства, в нем должны быть встроены средства, реализующие некоторый набор сетевых протоколов.

При организации взаимодействия могут быть использованы два основных типа протоколов. В протоколах *с установлением соединения* (connection-oriented network service, CONS) перед обменом данными отправитель и получатель должны сначала установить логическое соединение, то есть договориться о параметрах процедуры обмена, которые будут действовать только в рамках данного соединения. После завершения диалога они должны разорвать это соединение. Когда устанавливается новое соединение, переговорная процедура выполняется заново. Телефон - это пример взаимодействия, основанного на установлении соединения.

Вторая группа протоколов - протоколы *без предварительного установления* соединения (connectionless network service, CLNS). Такие протоколы называются также *дейтаграммными* протоколами. Отправитель просто передает сообщение, когда оно готово. Опускание письма в почтовый ящик - это пример связи без установления соединения.

Согласованный набор протоколов разных уровней, достаточный для организации межсетевого взаимодействия, называется *стеком протоколов*.

Существует достаточно много стеков протоколов, широко применяемых в сетях. Это и стеки, являющиеся международными и национальными стандартами, и фирменные стеки, получившие распространение благодаря распространенности оборудования той или иной фирмы. Примерами популярных стеков протоколов могут служить стек IPX/SPX фирмы Novell,

стек TCP/IP, используемый в сети Internet и во многих сетях на основе операционной системы UNIX, стек OSI международной организации по стандартизации, стек DECnet корпорации Digital Equipment и некоторые другие.

2.2.1 Стек OSI

Следует различать стек протоколов OSI и модель OSI. В то время как модель OSI концептуально определяет процедуру взаимодействия открытых систем, стек OSI - это набор вполне конкретных спецификаций протоколов, образующих согласованный стек протоколов. Это международный, независимый от производителей стандарт. По вполне очевидным причинам стек OSI в отличие от других стандартных стеков полностью соответствует модели взаимодействия OSI, он включает спецификации для всех семи уровней модели взаимодействия открытых систем (рисунок 26).

Рисунок 26 - Стек OSI

На физическом и канальном уровнях стек OSI поддерживает протоколы Ethernet, Token Ring, FDDI, а также протоколы LLC, X.25 и ISDN.

На сетевом уровне реализованы протоколы, как без установления соединений, так и с установлением соединений. Транспортный протокол стека OSI в соответствии с функциями, определенными для него в модели OSI, скрывает различия между сетевыми сервисами с установлением соединения и без установления соединения, так что пользователи получают нужное качество обслуживания независимо от нижележащего сетевого уровня. Чтобы обеспечить это, транспортный уровень требует, чтобы пользователь задал нужное качество обслуживания. Определены 5 классов транспортного сервиса, от низшего класса 0 до высшего класса 4, которые отличаются степенью

устойчивости к ошибкам и требованиями к восстановлению данных после ошибок.

Сервисы прикладного уровня включают передачу файлов, эмуляцию терминала, службу каталогов и почту. Из них наиболее перспективными являются служба каталогов (стандарт X.500), электронная почта (X.400), протокол виртуального терминала (VT), протокол передачи, доступа и управления файлами (FTAM), протокол пересылки и управления работами (JTM). В последнее время ISO сконцентрировала свои усилия именно на сервисах верхнего уровня.

X.400 - описывают модель системы обмена сообщениями, протоколы взаимодействия между всеми компонентами этой системы, а также множество видов сообщений и возможности, которыми обладает отправитель по каждому виду отправляемых сообщений.

Целью рекомендаций X.500 является выработка стандартов глобальной справочной службы. Процесс доставки сообщения требует знания адреса получателя, что при больших размерах сетей представляет собой проблему, поэтому необходимо иметь справочную службу, помогающую получать адреса отправителей и получателей. В общем виде служба X.500 представляет собой распределенную базу данных имен и адресов. Все пользователи потенциально имеют право войти в эту базу данных, используя определенный набор атрибутов.

Над базой данных имен и адресов определены следующие операции: чтение (получение адреса по известному имени), запрос (получение имени по известным атрибутам адреса), модификация, включающая удаление и добавление записей в базе данных.

Учет рекомендаций X.400 и X.500 при проектировании систем электронной почты делает принципиально возможной и концептуально простой стыковку почтовых систем разных производителей.

Протокол VT решает проблему несовместимости различных протоколов эмуляции терминалов. Сейчас пользователю персонального компьютера, совместимого с IBM PC, для одновременной работы с компьютерами VAX, IBM 3090 и HP9000 нужно приобрести три различные программы для эмуляции терминалов различных типов и использующих разные протоколы. Если бы каждый хост-компьютер имел бы в своем составе программное обеспечение протокола эмуляции терминала ISO, то и пользователю бы понадобилась только одна программа, поддерживающая протокол VT.

Передача файлов - это наиболее распространенный компьютерный сервис. ISO предусматривает такой сервис в протоколе FTAM, который предусматривает средства для локализации и доступа к содержимому файла и включает набор директив для вставки, замены, расширения и очистки содержимого файла. FTAM также предусматривает средства для манипулирования файлом как единым целым, включая создание, удаление, чтение, открытие, закрытие файла и выбор его атрибутов.

Протокол пересылки и управления работами JTM позволяет пользователям пересылать работы, которые должны быть выполнены на хост-

компьютере. Язык управления заданиями, который обеспечивает передачу работ, указывает хост-компьютеру, какие действия и с какими программами и файлами должны быть выполнены. Протокол JTM поддерживает традиционную пакетную обработку, обработку транзакций, ввод удаленных заданий и доступ к распределенным базам данных.

2.2.2 Стек TCP/IP

Стек TCP/IP, называемый также стеком Internet, является одним из наиболее популярных и перспективных стеков коммуникационных протоколов.

Так как стек TCP/IP был разработан до появления модели взаимодействия открытых систем ISO/OSI, то, хотя он также имеет многоуровневую структуру, соответствие уровней стека TCP/IP уровням модели OSI достаточно условно /8/.

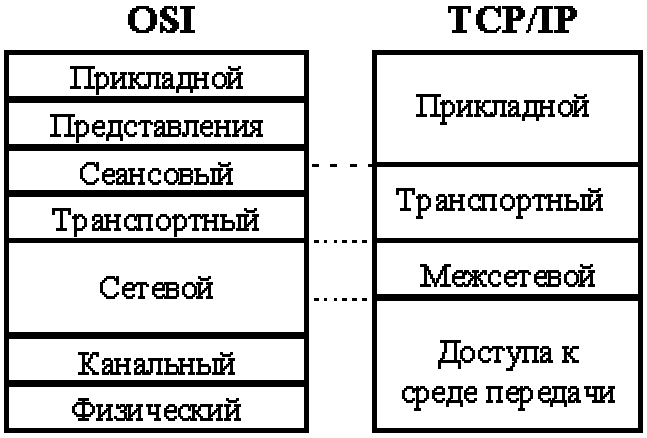


Рисунок 27 – Соответствие стека протоколов TCP/IP модели OSI

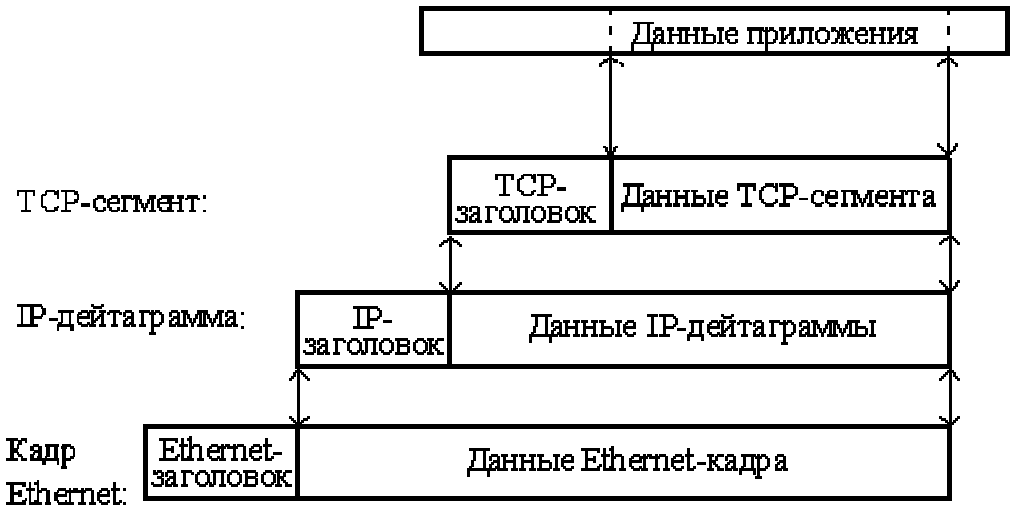


Рисунок 28 - Пример инкапсуляции пакетов в стеке TCP/IP

Как и в модели OSI, данные более верхних уровней инкапсулируются в пакеты нижних уровней (рисунок 28).

Стек протоколов TCP/IP делится на 4 уровня: *прикладной (application), транспортный (transport), межсетевой (internet) и уровень доступа к среде передачи (network access)*. Структура протоколов TCP/IP приведена на рисунке 29.

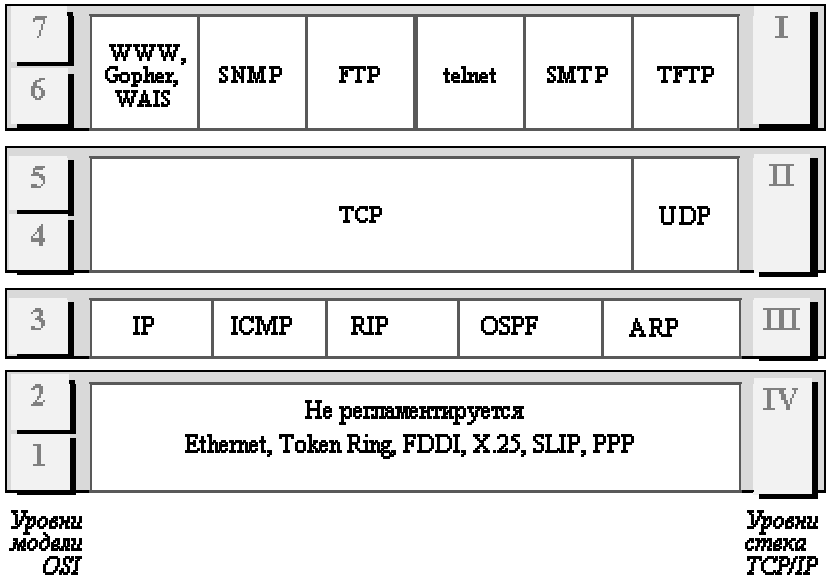


Рисунок 29 - Стек TCP / IP

Ниже кратко рассматриваются функции каждого уровня и примеры протоколов.

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

Следующий уровень (уровень III) - это уровень межсетевого взаимодействия, который занимается передачей дейтаграмм с использованием различных локальных сетей, территориальных сетей X.25, линий специальной связи и т. п. В качестве основного протокола сетевого уровня в стеке используется протокол IP, который изначально проектировался как протокол передачи пакетов в составных сетях, состоящих из большого количества локальных сетей, объединенных как локальными, так и глобальными связями. Поэтому протокол IP хорошо работает в сетях со сложной топологией,

рационально используя наличие в них подсистем и экономно расходуя пропускную способность низкоскоростных линий связи.

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

Следующий уровень (уровень II) называется транспортным. Протоколы транспортного уровня обеспечивают прозрачную (сквозную) доставку данных (end-to-end delivery service) между двумя прикладными процессами. Процесс, получающий или отправляющий данные с помощью транспортного уровня, идентифицируется на этом уровне номером, который называется номером порта. Таким образом, роль адреса отправителя и получателя на транспортном уровне выполняет номер порта (или проще - порт).

Анализируя заголовок своего пакета, полученного от межсетевого уровня, транспортный модуль определяет по номеру порта получателя, какому из прикладных процессов направлены данные, и передает эти данные соответствующему прикладному процессу (возможно, после проверки их на наличие ошибок и т.п.). Номера портов получателя и отправителя записываются в заголовок транспортным модулем, отправляющим данные; заголовок транспортного уровня содержит также и другую служебную информацию; формат заголовка зависит от используемого транспортного протокола.

На транспортном уровне работают два основных протокола: UDP(User Datagram Protocol) и TCP (Transmission Control Protocol).

TCP (Transmission Control Protocol - протокол контроля передачи) - надежный протокол с установлением соединения: он управляет логическим сеансом связи (устанавливает, поддерживает и закрывает соединение) между процессами и обеспечивает надежную (безошибочную и гарантированную) доставку прикладных данных от процесса к процессу.

Протокол UDP обеспечивает передачу прикладных пакетов дейтаграммным методом, то есть без установления виртуального соединения, и поэтому требует меньших накладных расходов, чем TCP.

Данными для TCP является не интерпретируемая протоколом последовательность пользовательских октетов, разбиваемая для передачи по частям. Каждая часть передается в отдельном TCP-сегменте. Для продвижения сегмента по сети между компьютером-отправителем и компьютером-получателем модуль TCP пользуется сервисом межсетевого уровня (вызывает модуль IP).

Самый верхний уровень (уровень I) называется прикладным уровнем. За долгие годы использования в сетях различных стран и организаций стек TCP/IP накопил большое количество протоколов и сервисов прикладного уровня. Для пересылки данных другому приложению, приложение обращается к тому или иному модулю транспортного уровня.

Протокол SNMP (Simple Network Management Protocol) используется для организации сетевого управления. Проблема управления разделяется здесь на две задачи. Первая задача связана с передачей информации. Протоколы передачи управляющей информации определяют процедуру взаимодействия сервера с программой-клиентом, работающей на хосте администратора. Они определяют форматы сообщений, которыми обмениваются клиенты и серверы, а также форматы имен и адресов. Вторая задача связана с контролируемыми данными. Стандарты регламентируют, какие данные должны сохраняться и накапливаться в шлюзах, имена этих данных и синтаксис этих имен.

Протокол пересылки файлов FTP (File Transfer Protocol) реализует удаленный доступ к файлу. Для того, чтобы обеспечить надежную передачу, FTP использует в качестве транспорта протокол с установлением соединений - TCP. Кроме пересылки файлов протокол, FTP предлагает и другие услуги. Так пользователю предоставляется возможность интерактивной работы с удаленной машиной, например, он может распечатать содержимое ее каталогов, FTP позволяет пользователю указывать тип и формат запоминаемых данных. Наконец, FTP выполняет аутентификацию пользователей. Прежде, чем получить доступ к файлу, в соответствии с протоколом пользователи должны сообщить свое имя и пароль.

Приложения, которым не требуются все возможности FTP, могут использовать другой, более экономичный протокол - простейший протокол пересылки файлов TFTP (Trivial File Transfer Protocol). Этот протокол реализует только передачу файлов, причем в качестве транспорта используется более простой, чем TCP, протокол без установления соединения - UDP.

Протокол telnet обеспечивает передачу потока байтов между процессами, а также между процессом и терминалом. Наиболее часто этот протокол используется для эмуляции терминала удаленной ЭВМ. При использовании сервиса telnet пользователь фактически управляет удаленным компьютером так же, как и локальный пользователь, поэтому такой вид доступа требует хорошей защиты.

2.2.3 Стек IPX/SPX

Этот стек является оригинальным стеком протоколов фирмы Novell, который она разработала для своей сетевой операционной системы NetWare еще в начале 80-х годов. Протоколы Internetwork Packet Exchange (IPX) и Sequenced Packet Exchange (SPX), которые дали имя стеку, являются прямой адаптацией протоколов XNS фирмы Xerox.

Семейство протоколов фирмы Novell и их соответствие модели ISO/OSI представлено на рисунке 30.

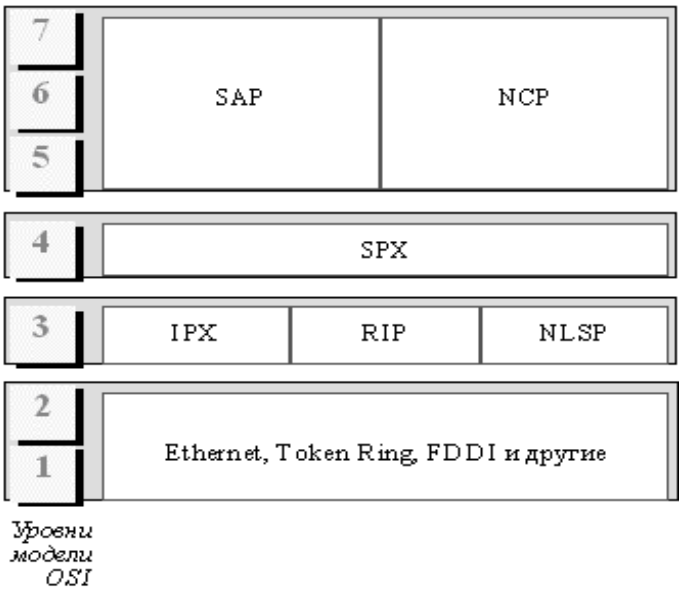


Рисунок 30 - Стек IPX / SPX

На физическом и канальном уровнях в сетях Novell используются все популярные протоколы этих уровней (Ethernet, Token Ring, FDDI и другие).

На сетевом уровне в стеке Novell работает протокол IPX, а также протоколы обмена маршрутной информацией RIP и NLSP (аналог протокола OSPF стека TCP/IP). IPX является протоколом, который занимается вопросами адресации и маршрутизации пакетов в сетях Novell. Маршрутные решения IPX основаны на адресных полях в заголовке его пакета, а также на информации, поступающей от протоколов обмена маршрутной информацией. Например, IPX использует информацию, поставляемую либо протоколом RIP, либо протоколом NLSP для передачи пакетов компьютеру назначения или следующему маршрутизатору. Протокол IPX обеспечивает выполнение трех функций: задание адреса, установление маршрута и рассылку дейтаграмм.

Транспортному уровню модели OSI в стеке Novell соответствует протокол SPX, который осуществляет передачу сообщений с установлением соединений.

На верхних прикладном, представительном и сеансовом уровнях работают протоколы NCP и SAP. Протокол NCP (NetWare Core Protocol) является протоколом взаимодействия сервера NetWare и оболочки рабочей станции. Этот протокол прикладного уровня реализует архитектуру клиент-сервер на верхних уровнях модели OSI. С помощью функций этого протокола рабочая станция производит подключение к серверу, отображает каталоги сервера на локальные буквы дисководов, просматривает файловую систему сервера, копирует удаленные файлы, изменяет их атрибуты и т.п., а также осуществляет разделение сетевого принтера между рабочими станциями.

SAP (Service Advertising Protocol) - протокол объявления о сервисе - концептуально подобен протоколу RIP. Подобно тому, как протокол RIP позволяет маршрутизаторам обмениваться маршрутной информацией, протокол

SAP дает возможность сетевым устройствам обмениваться информацией об имеющихся сетевых сервисах.

Особенности стека IPX/SPX обусловлены особенностями ОС NetWare, а именно ориентацией ее ранних версий (до 4.0) на работу в локальных сетях небольших размеров, состоящих из персональных компьютеров со скромными ресурсами. Поэтому Novell нужны были протоколы, на реализацию которых требовалось минимальное количество оперативной памяти (ограниченной в IBM-совместимых компьютерах под управлением MS-DOS 640 Кбайтами) и которые бы быстро работали на процессорах небольшой вычислительной мощности. В результате, протоколы стека IPX/SPX до недавнего времени хорошо работали в локальных сетях и не очень - в больших корпоративных сетях, так как слишком перегружали медленные глобальные связи широковещательными пакетами. Однако к моменту выпуска версии NetWare 4.0, Novell внесла в свои протоколы серьезные изменения, направленные на приспособление их для работы в корпоративных сетях. Сейчас стек IPX/SPX реализован не только в NetWare, но и в нескольких других популярных сетевых ОС - SCO UNIX, Sun Solaris, Microsoft Windows NT.

2.3 Контрольные вопросы

1) Эталонная модель взаимодействия открытых систем.
2) Особенности физического, канального, сетевого, транспортного, сеансового, прикладного и представительского уровней эталонной модели OSI.
3) Стек протоколов OSI.
4) Стек протоколов TCP/IP.
5) Стек протоколов IPX/SPX.

2.4 Тесты

1) Укажите правильный порядок следования наименования уровней в модели OSI. Обозначения: S – сеансовый, N – сетевой, PH – физический, P – представительный, D – канальный, T – транспортный, A – уровень приложений.
   а) S, N, PH, P, D, T, A;
   б) A, S, T, P, N, D, PH;
   в) A, S, T, P, N, D, PH;
   г) A, P, S, T, N, D, PH.

2) Укажите наименование блока данных сетевого уровня:
   а) кадр;
   б) сегмент;
   в) пакет;
   г) сообщение.

3) Какая из перечисленных функций не реализуется на сеансовом уровне OSI?

а) установление сессии;
б) разрывание сессии;
в) обслуживание двунаправленного обмена сообщениями;
г) обнаружение сегментов, которые содержат ошибки.

4) Объекты какого уровня модели OSI обеспечивают доставку данных от источника до приемника?
а) канальный;
б) сеансовый;
в) транспортный;
г) сетевой.

5) Какие из 2 перечисленных функций не выполняются объектами представительного уровня?
а) шифрование;
б) сжатие данных;
в) опознавание;

6) Укажите устройства, которые реализуют функции физического уровня модели OSI?
а) хаб;
б) мост;
в) репитер;
г) коммутатор;

7) Совокупность алгоритмов взаимодействия объектов одноименных уровней определяет понятие:
а) интерфейс;
б) стек;
в) протокол;
г) уровень.

8) Укажите 2 обязательных компонента сетевого адреса?
а) физический адрес;
б) адрес сети;
в) адрес хоста;
г) адрес порта.

9) На каком уровне модели OSI коммутатор выполняет обработку данных?
а) канальный;
б) транспортный;
в) физический;
г) сетевой.

10) На каком уровне модели OSI производится преобразование данных в поток бит?

а) канальный;
б) транспортный;
в) представительный;
г) физический.

11) Укажите уровни OSI, на которых выполняется инкапсуляция?

а) физический;
б) сеансовый;
в) транспортный;
г) представительный.

12) Укажите устройство, которое реализует функции сетевого уровня модели OSI?

а) маршрутизатор;
б) репитер;
в) коммутатор;
г) хаб.

13) Какие из перечисленных функций не реализуются протоколами сетевого уровня?

а) определение маршрута;
б) обеспечение доставки данных в том порядке, в каком они были переданы;
в) определение логического адреса;
г) управление потоком.

14) Какие из перечисленных протоколов являются протоколами физического уровня?

а) V.24;
б) MPEG3;
в) HSSI;
г) ASCII.

15) На каком уровне OSI определяется физический адрес объекта?

а) сетевой;
б) физический;
в) транспортный;
г) канальный.

3 Объединение сетей с помощью мостов, коммутаторов и маршрутизаторов

3.1 Устройства объединения сетей

Устройства объединения сетей обеспечивают связь между сегментами локальных сетей, отдельными ЛВС и подсетями любого уровня. Эти устройства в самом общем виде могут быть отнесены к определенным уровням эталонной модели взаимодействия открытых систем.

Соотношение между функциями этих устройств и уровнями модели OSI показано на рисунке 31.

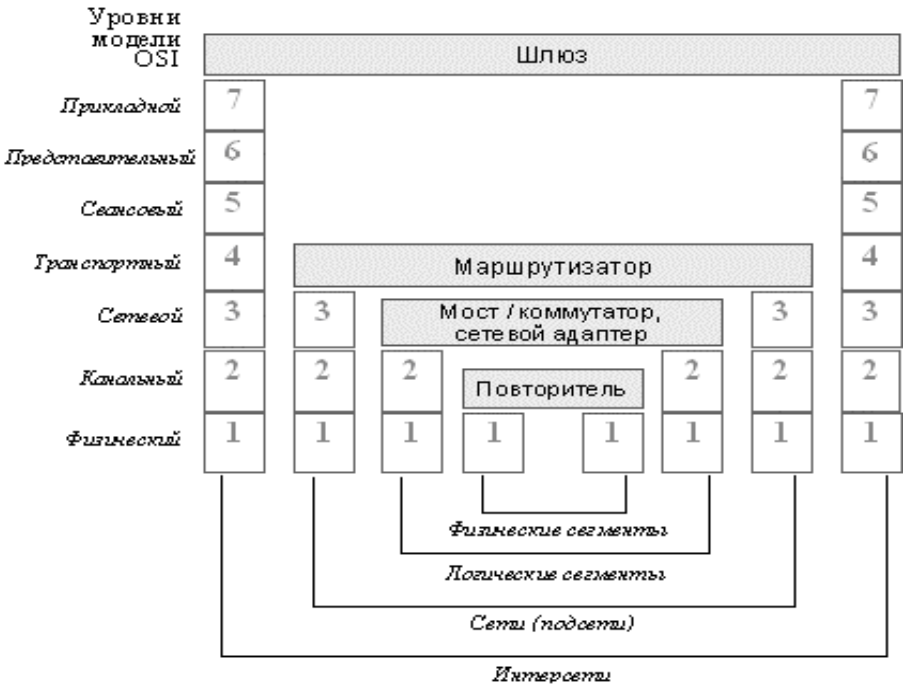


Рисунок 31 - Соответствие функций коммуникационного оборудования модели OSI

Существуют следующие классы устройств для объединения сегментов и сетей. Повторитель, который регенерирует сигналы, за счет чего позволяет увеличивать длину сети, работает на физическом уровне.

Сетевой адаптер также работает на физическом и отчасти на канальном уровнях. К физическому уровню относится та часть функций сетевого адаптера, которая связана с приемом и передачей сигналов по линии связи, а получение доступа к разделяемой среде передачи, распознавание MAC-адреса компьютера - это уже функция канального уровня.

Мосты (bridges) и коммутаторы (switches) объединяют сети на канальном уровне и используют функциональные возможности физического уровня. Мосты выполняются на основе компьютера, оснащенного соответствующим ПО. Отличие коммутаторов от мостов в том, что они реализуют свои функции аппаратными средствами и поэтому обладают значительно более высоким

быстродействием;

Для мостов сеть представляется набором MAC-адресов устройств. Они извлекают эти адреса из заголовков, добавленных к пакетам на канальном уровне, и используют их во время обработки пакетов для принятия решения о том, на какой порт отправить тот или иной пакет. Мосты не имеют доступа к информации об адресах сетей, относящейся к более высокому уровню. Поэтому они ограничены в принятии решений о возможных путях или маршрутах перемещения пакетов по сети /9/.

Маршрутизаторы работают на сетевом уровне модели OSI. Для маршрутизаторов сеть - это набор сетевых адресов устройств и множество сетевых путей. Маршрутизаторы анализируют все возможные пути между любыми двумя узлами сети и выбирают самый короткий из них.

На рисунке 32 показан еще один тип коммуникационных устройств - шлюз, который может работать на любом уровне модели OSI. *Шлюз* (gateway) - это устройство, выполняющее трансляцию протоколов. Шлюз размещается между взаимодействующими сетями и служит посредником, переводящим сообщения, поступающие из одной сети, в формат другой сети. Шлюз может быть реализован как чисто программными средствами, установленными на обычном компьютере, так и на базе специализированного компьютера.

Фрагмент вычислительной сети (рисунок 32) включает основные типы коммуникационного оборудования, для образования локальных сетей и соединения их через глобальные связи друг с другом.

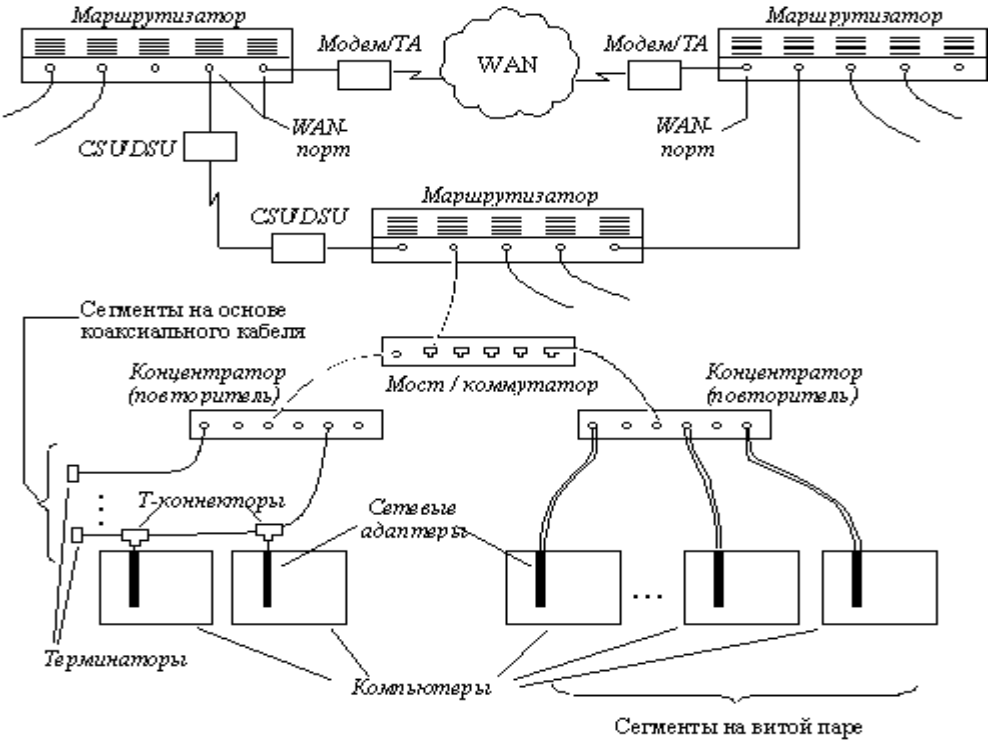


Рисунок 32 - Фрагмент сети

Для подключения локальных сетей к глобальным связям используются специальные выходы (WAN-порты) мостов и маршрутизаторов, а также аппаратура передачи данных по длинным линиям - модемы (при работе по

аналоговым линиям) или же устройства подключения к цифровым каналам (ТА - терминальные адаптеры сетей ISDN, устройства обслуживания цифровых выделенных каналов типа CSU/DSU и т.п.).

3.2 Физическая структуризация локальной сети. Повторители и концентраторы

Для построения простейшей односегментной сети достаточно иметь сетевые адаптеры и кабель подходящего типа. Но даже в этом простом случае часто используются дополнительные устройства - повторители сигналов, позволяющие преодолеть ограничения на максимальную длину кабельного сегмента.

Основная функция *повторителя* (repeater), как это следует из его названия - повторение сигналов, поступающих на один из его портов, на всех остальных портах (Ethernet) или на следующем в логическом кольце порте (Token Ring, FDDI) синхронно с сигналами-оригиналами. Повторитель улучшает электрические характеристики сигналов и их синхронность, и за счет этого появляется возможность увеличивать общую длину кабеля между самыми удаленными в сети станциями.

Многопортовый повторитель часто называют *концентратором* (hub, concentrator), что отражает тот факт, что данное устройство реализует не только функцию повторения сигналов, но и концентрирует в одном центральном устройстве функции объединения компьютеров в сеть. Практически во всех современных сетевых стандартах концентратор является необходимым элементом сети, соединяющим отдельные компьютеры в сеть.

Отрезки кабеля, соединяющие два компьютера или какие либо два других сетевых устройства называются *физическими сегментам*. Таким образом, концентраторы и повторители, которые используются для добавления новых физических сегментов, являются средством физической структуризации сети.

Основной задачей повторителя является восстановление электрических сигналов для передачи их в другие сегменты. За счет усиления и восстановления формы электрических сигналов повторителем становится возможным расширение сетей, построенных на основе коаксиального кабеля и увеличение общего числа пользователей сети.

Концентраторы образуют из отдельных физических отрезков кабеля общую среду передачи данных - *логический сегмент*. Логический сегмент также называют доменом коллизий, поскольку при попытке одновременной передачи данных любых двух компьютеров этого сегмента, хотя бы и принадлежащих разным физическим сегментам, возникает блокировка передающей среды. Появление устройств, централизующих соединения между отдельными сетевыми устройствами, потенциально позволяет улучшить управляемость сети и ее эксплуатационные характеристики (модифицируемость, ремонтопригодность и т.п.). С этой целью разработчики концентраторов часто встраивают в свои устройства, кроме основной функции

повторителя, ряд вспомогательных функций, весьма полезных для улучшения качества сети, наиболее часто встречаются следующие:

– объединение сегментов с различными физическими средами (например, коаксиал, витая пара и оптоволокно) в единый логический сегмент;

– автосегментация портов - автоматическое отключение порта при его некорректном поведении (повреждение кабеля, интенсивная генерация пакетов ошибочной длины и т.п.);

– поддержка между концентраторами резервных связей, которые используются при отказе основных;

– защита передаваемых по сети данных от несанкционированного доступа (например, путем искажения поля данных в кадрах, повторяемых на портах, не содержащих компьютера с адресом назначения).

Сегодняшние повторители часто позволяют организовать несколько сегментов (рисунок 33), каждый из которых предоставляет пользователям отдельную разделяемую полосу 10Мбит/с /10/.

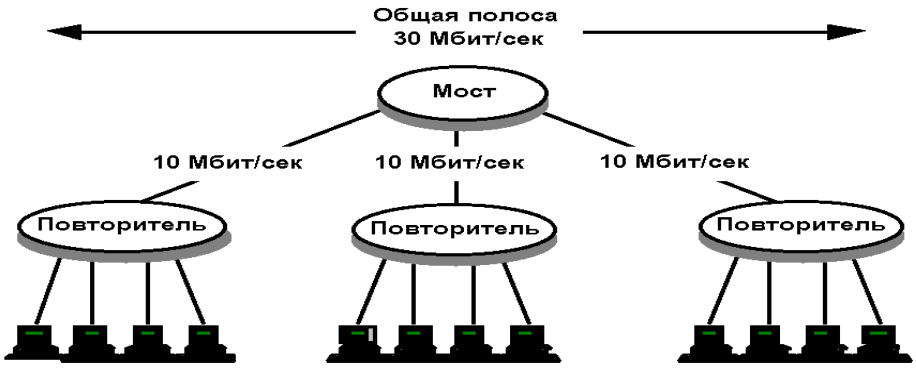


Рисунок 33 - Сегментация сети. Переключение портов

Некоторые концентраторы позволяют программным путем разделять порты устройства на независимые сегменты - такая возможность называется переключением портов. Концентратор, к примеру, может содержать три различных сегмента Ethernet, организуемые внутренними средствами. Переключение портов обеспечивает администратору сети высокую гибкость организации сегментов, позволяя переносить порты из одного сегмента в другой программными средствами. Эта возможность особенно полезна для распределения нагрузки между сегментами Ethernet и снижения расходов.

3.3 Логическая структуризация сети. Мосты и коммутаторы

Несмотря на появление новых дополнительных возможностей, основной функцией концентраторов остается передача пакетов по общей разделяемой среде. Коллективное использование многими компьютерами общей кабельной системы в режиме разделения времени приводит к существенному снижению производительности сети при интенсивном трафике. Общая среда перестает

справляться с потоком передаваемых кадров и в сети возникает очередь компьютеров, ожидающих доступа.

Поэтому сети, построенные на основе концентраторов, не могут расширяться в требуемых пределах - при определенном количестве компьютеров в сети или при появлении новых приложений всегда происходит насыщение передающей среды, и задержки в ее работе становятся недопустимыми. Эта проблема может быть решена путем логической структуризации сети с помощью мостов, коммутаторов и маршрутизаторов.

Мосты имеют много отличий от повторителей. Повторители передают все пакеты, а мосты - только те, которые нужно. Если пакет не нужно передавать в другой сегмент, он фильтруется. Для мостов существуют многочисленные алгоритмы (правила) передачи и фильтрации пакетов - минимальным требованием является фильтрация пакетов по адресу получателя.

Другим важным отличием мостов от повторителей является то, что сегменты, подключенные к повторителю, образуют одну разделяемую среду, а сегменты, подключенные к каждому порту моста образуют свою среду с полосой 10Мбит/с.

При использовании моста пользователи одного сегмента разделяют полосу, а пользователи разных сегментов - используют независимые среды. Следовательно, мост обеспечивает преимущества как с точки зрения расширения сети, так и обеспечения большей полосы для каждого пользователя (рисунок 34).

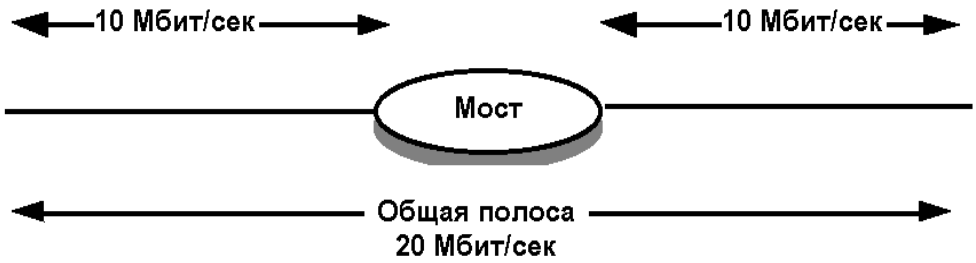


Рисунок 34 - Мост

Мосты выполняют несложные функции: анализируют поступающие фреймы и, базируясь на информации, содержащейся во фреймах, принимают решения о их пересылке к месту назначения.

Поскольку мосты функционируют на канальном уровне, они могут быстро продвигать трафик, представляющий любой протокол сетевого уровня, не проверяя информацию высших уровней. Это свойство прозрачности мостов для протоколов верхних уровней позволяет, например, продвигать трафик протоколов Apple Talk, DECnet, TCP/IP, XNS и других между двумя и более сетями и является основным преимуществом использования мостов для создания объединенных сетей.

Таким образом, достоинства использования мостов:

- мосты увеличивают число связанных сетью устройств и эффективную

длину ЛВС, позволяя подключать дополнительные отдаленные станции и сетевые сегменты;

- разделяя крупные сети на автономные блоки, мосты уменьшают трафик в отдельных сегментах и создают преграду для распространения некоторых потенциально опасных для сети неисправностей.

Можно выделить два основных типа мостов:

- локальные мосты обеспечивают прямое соединение множества сегментов ЛВС, находящихся на одной территории;
- дистанционные мосты соединяют множество сегментов ЛВС на различных территориях, обычно через телекоммуникационные линии /11/.

Мост (bridge), а также его быстродействующий функциональный аналог - коммутатор (switching hub), делит общую среду передачи данных на логические сегменты. Логический сегмент образуется путем объединения нескольких физических сегментов (отрезков кабеля) с помощью одного или нескольких концентраторов. Каждый логический сегмент подключается к отдельному порту моста/коммутатора (рисунок 35). При поступлении кадра на какой-либо из портов мост/коммутатор повторяет этот кадр, но не на всех портах, как это делает концентратор, а только на том порту, к которому подключен сегмент, содержащий компьютер-адресат.

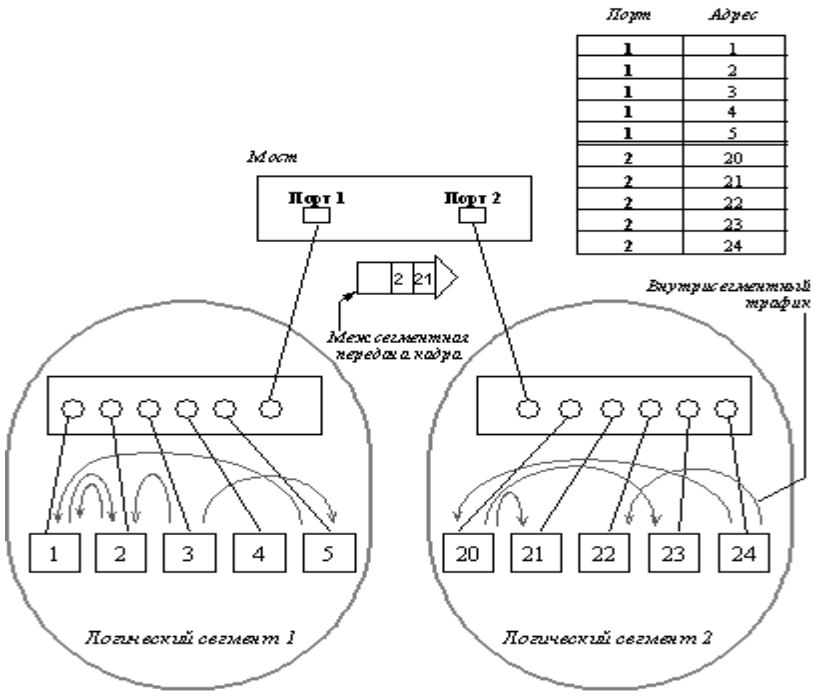


Рисунок 35 - Разделение сети на логические сегменты

Разница между мостом и коммутатором состоит в том, что мост в каждый момент времени может осуществлять передачу кадров только между одной парой портов, а коммутатор одновременно поддерживает потоки данных между всеми своими портами. Другими словами, мост передает кадры последовательно, а коммутатор параллельно. (Для упрощения изложения далее в этом разделе будет использоваться термин "коммутатор" для обозначения

этих обоих разновидностей устройств, поскольку все сказанное ниже в равной степени относится и к мостам, и к коммутаторам.) Следует отметить, что в последнее время локальные мосты полностью вытеснены коммутаторами. Мосты используются только для связи локальных сетей с глобальными, то есть как средства удаленного доступа, поскольку в этом случае необходимость в параллельной передаче между несколькими парами портов просто не возникает.

При работе коммутатора среда передачи данных каждого логического сегмента остается общей только для тех компьютеров, которые подключены к этому сегменту непосредственно. Коммутатор осуществляет связь сред передачи данных различных логических сегментов. Он передает кадры между логическими сегментами только при необходимости, то есть только тогда, когда взаимодействующие компьютеры находятся в разных сегментах.

Деление сети на логические сегменты улучшает производительность сети, если в сети имеются группы компьютеров, преимущественно обменивающиеся информацией между собой.

Коммутаторы принимают решение о том, на какой порт нужно передать кадр, анализируя адрес назначения, помещенный в кадре, а также на основании информации о принадлежности того или иного компьютера определенному сегменту, подключенному к одному из портов коммутатора, то есть на основании информации о конфигурации сети. Для того, чтобы собрать и обработать информацию о конфигурации подключенных к нему сегментов, коммутатор должен пройти стадию "обучения", то есть самостоятельно проделать некоторую предварительную работу по изучению проходящего через него трафика. Определение принадлежности компьютеров сегментам возможно за счет наличия в кадре не только адреса назначения, но и адреса источника, сгенерировавшего пакет. Используя информацию об адресе источника, коммутатор устанавливает соответствие между номерами портов и адресами компьютеров. В процессе изучения сети коммутатор просто передает появляющиеся на входах его портов кадры на все остальные порты, работая некоторое время повторителем. После того, как коммутатор узнает о принадлежности адресов сегментам, он начинает передавать кадры между портами только в случае межсегментной передачи. Если, уже после завершения обучения, на входе коммутатора вдруг появится кадр с неизвестным адресом назначения, то этот кадр будет повторен на всех портах.

3.3.1 Как работает коммутатор

Коммутатор - это сетевое устройство, обеспечивающее передачу информации от порта источника информации к порту назначения с минимальными задержками и низкими накладными расходами. Ядром коммутатора является коммутационная матрица, обеспечивающая передачу данных между любыми двумя точками, или быстродействующая шина, через которую любой порт может передать информацию любому другому порту. Кроме того, в коммутаторе должны быть адаптеры портов, позволяющие

преобразовывать протокол, который использует присоединенное к порту устройство, в форму, приемлемую для ядра коммутатора.

Можно сказать, что в ядре коммутатора образуется нечто вроде сверхбыстрой виртуальной сети. Источник данных передает пакеты локальной сети в адаптер порта, к которому он подключен, а адаптер преобразует адрес назначения пакета в адрес порта коммутатора. Затем адаптер передает пакет в ядро коммутатора, добавив к нему вычисленный адрес порта. Получив многоадресное сообщение, коммутатор направляет его всем участникам виртуальной сети источника сообщения. При этом предполагается, что сеть работает настолько быстро, что суммарный объем трафика не может превысить ее пропускной способности.

Коммутатор Ethernet поддерживает внутреннюю таблицу, связывающую порты с адресами подключенных к ним устройств (таблица 5). Эту таблицу администратор сети может создать самостоятельно или задать ее автоматическое создание средствами коммутатора.

Используя таблицу адресов и содержащийся в пакете адрес получателя, коммутатор организует виртуальное соединение порта отправителя с портом получателя и передает пакет через это соединение.

Таблица 5 – Связь портов с адресами подключенных к ним устройств

MAC-адрес	Номер порта
A	1
B	2
C	3
D	4

На рисунке 36 узел A посылает пакет узлу D. Найдя адрес получателя в своей внутренней таблице, коммутатор передает пакет в порт 4.

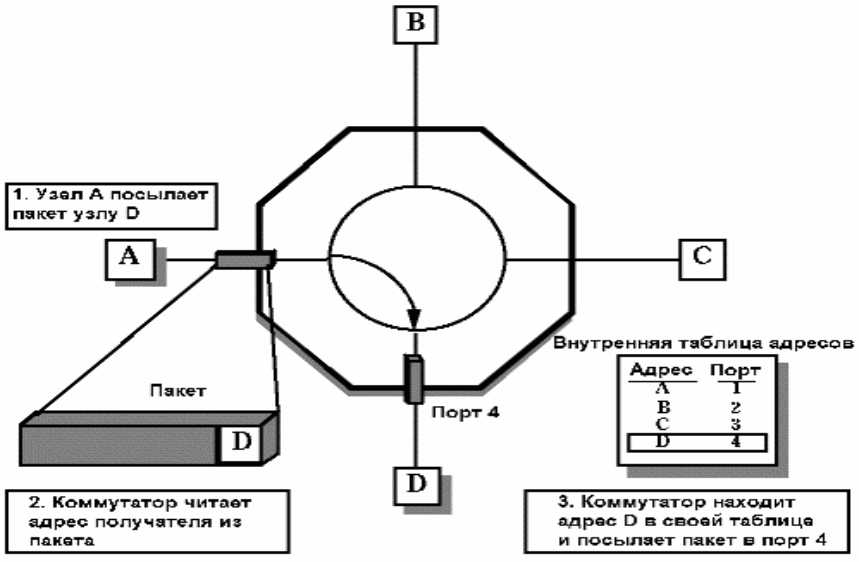


Рисунок 36 - Как работает коммутатор

Виртуальное соединение между портами коммутатора сохраняется в течение передачи одного пакета, то есть для каждого пакета виртуальное соединение организуется заново на основе содержащегося в этом пакете адреса получателя. Поскольку пакет передается только в тот порт, к которому подключен адресат, остальные пользователи (в нашем примере - B и C) не получат этот пакет. Таким образом, коммутаторы обеспечивают средства безопасности, недоступные для стандартных повторителей.

В коммутаторах Ethernet передача данных между любыми парами портов происходит независимо и, следовательно, для каждого виртуального соединения выделяется вся полоса канала.

Например, коммутатор 10Мбит/с на рисунке 37 обеспечивает одновременную передачу пакета из A в D и из порта B в порт C с полосой 10Мбит/с для каждого соединения.

Поскольку для каждого соединения предоставляется вся полоса, суммарная пропускная способность коммутатора в приведенном примере составляет 20Мбит/с. Если данные передаются между большим числом пар портов, интегральная полоса соответственно расширяется. Например, 24 портовый коммутатор Ethernet может обеспечивать интегральную пропускную способность до 120Мбит/с при одновременной организации 12 соединений с полосой 10Мбит/с для каждого из них. Теоретически, интегральная полоса коммутатора растет пропорционально числу портов.

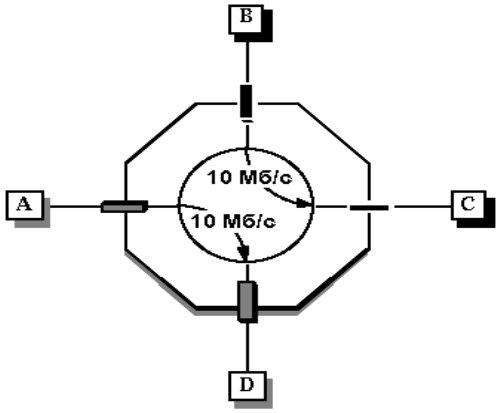


Рисунок 37 - Одновременные соединения

Коммутатор Ethernet 10Мбит/с может обеспечить высокую пропускную способность при условии организации одновременных соединений между всеми парами портов. Однако в реальной жизни трафик обычно представляет собой ситуацию "один ко многим" (например, множество пользователей сети обращается к ресурсам одного сервера). В таких случаях пропускная способность коммутатора в нашем примере не будет превышать 10 Мбит/с, и коммутатор не обеспечит существенного преимущества по сравнению с обычным концентратором (повторителем) /12/.

На рисунке 38 три узла A, B и D передают данные узлу C. Коммутатор сохраняет пакеты от узлов A и B в своей памяти до тех пор, пока не завершится передача пакета из узла D. После завершения передачи пакета коммутатор начинает передавать хранящиеся в памяти пакеты от узлов A и B. В данном

случае пропускная способность коммутатора определяется полосой канала С (в данном случае 10Мбит/с).

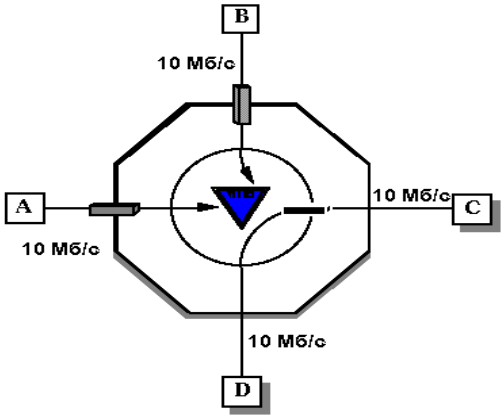


Рисунок 38 - Соединение "один ко многим"

Классы коммутаторов Ethernet. Хотя все коммутаторы имеют много общего, целесообразно разделить их на два класса, предназначенных для решения разных задач.

Коммутаторы для рабочих групп. Коммутаторы для рабочих групп обеспечивают выделенную полосу при соединении любой пары узлов, подключенных к портам коммутатора. Коммутатор обеспечивает для каждого порта выделенную полосу 10Мбит/с. Каждый порт коммутатора связан с уникальным адресом подключенного к данному порту устройства Ethernet.

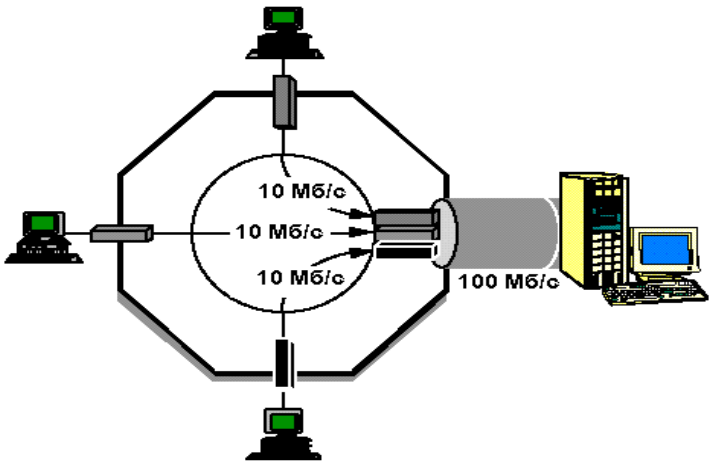


Рисунок 39 - Коммутатор в сети с архитектурой клиент-сервер

Коммутаторы рабочих групп могут работать со скоростью 10 или 100Мбит/с для различных портов. Такая возможность снижает уровень блокировки при попытке организации нескольких соединений клиентов 10Мбит/с с одним скоростным портом. В рабочих группах с архитектурой клиент-сервер несколько клиентов 10Мбит/с могут обращаться к серверу, подключенному к порту 100Мбит/с. В примере три узла 10Мбит/с

одновременно обращаются к серверу через порт 100Мбит/с. Из полосы 100Мбит/с, доступной для доступа к серверу, используется 30Мбит/с, а 70Мбит/с доступно для одновременного подключения к серверу еще семи устройств 10Мбит/с через виртуальные каналы.

Поддержка различных скоростей полезна также для объединения групповых коммутаторов Ethernet с использованием концентраторов 100Мбит/с Fast Ethernet (100Base-T) в качестве локальных магистралей. В показанной на рисунке 40 конфигурации, коммутаторы, поддерживающие скорости 10Мбит/с и 100Мбит/с подключены к повторителю 100Мбит/с.

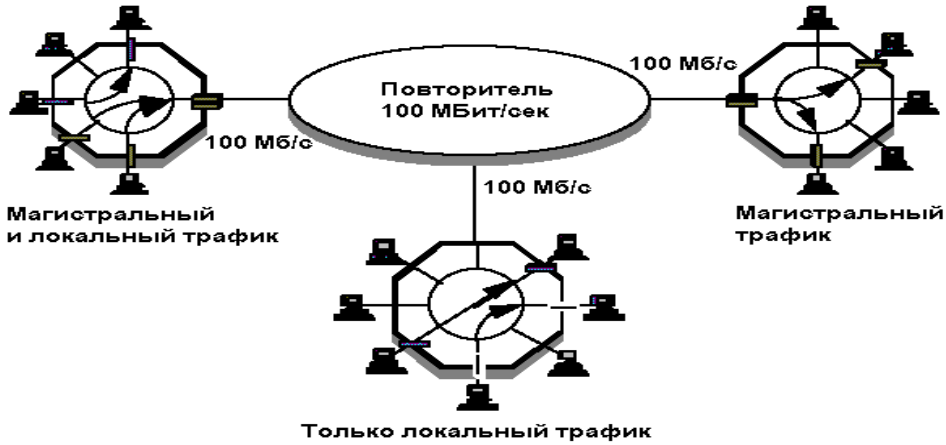


Рисунок 40 - Объединение коммутаторов для рабочих групп

Локальный трафик остается в пределах рабочей группы, а остальной трафик - передается в сеть через концентратор 100Мбит/с Ethernet. Для подключения к повторителю 10 или 100Мбит/с коммутатор должен иметь порт, способный работать с большим числом адресов Ethernet.

Основным преимуществом коммутаторов для рабочих групп является высокая производительность сети на уровне рабочей группы за счет предоставления каждому пользователю выделенной полосы канала (10Мбит/с). Кроме того, коммутаторы снижают (в пределе до нуля) количество коллизий - в отличие от магистральных коммутаторов, коммутаторы рабочих групп, не будут передавать коллизионные фрагменты адресатам. Коммутаторы для рабочих групп позволяют полностью сохранить сетевую инфраструктуру со стороны клиентов, включая программы, сетевые адаптеры, кабели. Стоимость коммутаторов для рабочих групп в расчете на один порт сегодня сравнима с ценами портов управляемых концентраторов.

Магистральные коммутаторы. Магистральные коммутаторы обеспечивают соединение со скоростью передачи среды между парой незанятых сегментов Ethernet. Если скорость портов для отправителя и получателя совпадают, сегмент получателя должен быть свободен во избежание блокировки.

На уровне рабочей группы каждый узел разделяет полосу 10Мбит/с с другими узлами в том же сегменте. Пакет, адресованный за пределы данной группы, будет передан магистральным коммутатором, как показано на рисунке 41. Магистральный коммутатор обеспечивает одновременную передачу пакетов

со скоростью среды между любыми парами своих портов. Подобно коммутаторам для рабочих групп, магистральные коммутаторы могут поддерживать различную скорость для своих портов. В большинстве случаев использование магистральных коммутаторов обеспечивает более простой и эффективный способ повышения производительности сети по сравнению с маршрутизаторами и мостами.

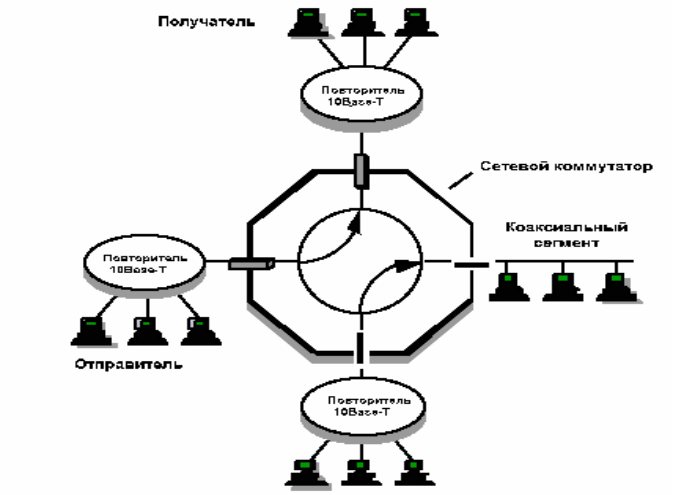


Рисунок 41 - Магистральный коммутатор

Основным недостатком при работе с магистральными коммутаторами является то, что на уровне рабочих групп пользователи работают с разделяемой средой, если они подключены к сегментам, организованным на основе повторителей или коаксиального кабеля. Более того, время отклика на уровне рабочей группы может быть достаточно большим. На уровне рабочей группы по-прежнему сохраняются коллизии, а фрагменты пакетов с ошибками будут пересылаться во все сети, подключенные к магистрали. Перечисленных недостатков можно избежать, если на уровне рабочих групп использовать коммутаторы взамен концентраторов (рисунок 42).

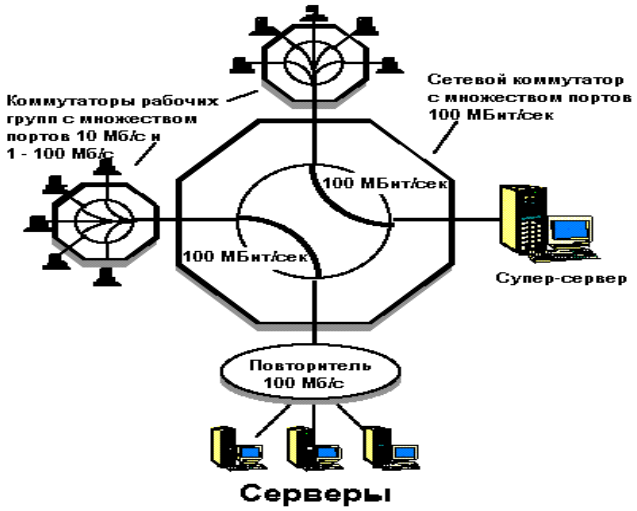


Рисунок 42 - Использование коммутаторов на уровне рабочих групп вместо концентраторов

В большинстве ресурсоемких приложений коммутатор 100Мбит/с может выполнять роль скоростной магистрали для коммутаторов рабочих групп с портами 10 и 100Мбит/с, концентраторами 100Мбит/с и серверами, в которых установлены адаптеры Ethernet 100Мбит/с.

3.4 Маршрутизаторы

Согласно определению крупнейшего производителя маршрутизаторов компании Cisco, маршрутизатор - "это устройство третьего уровня, использующее одну и более метрик для определения оптимального пути передачи сетевого трафика на основе информации сетевого уровня". По существу маршрутизатор представляет собой "компьютер" с необходимым программным обеспечением и устройствами ввода/вывода.

Маршрутизатор позволяет организовывать в сети избыточные связи, образующие петли. Он справляется с этой задачей за счет того, что принимает решение о передаче пакетов на основании более полной информации о графе связей в сети, чем мост или коммутатор. Маршрутизатор имеет в своем распоряжении базу топологической информации, которая говорит ему, например, о том, между какими подсетями общей сети имеются связи и в каком состоянии (работоспособном или нет) они находятся. Имея такую карту сети, маршрутизатор может выбрать один из нескольких возможных маршрутов доставки пакета адресату. В данном случае под маршрутом понимают последовательность прохождения пакетом маршрутизаторов. Например, на рисунке 50 для связи станций L2 сети LAN1 и L1 сети LAN6 имеется два маршрута: M1-M5-M7 и M1-M6-M7.

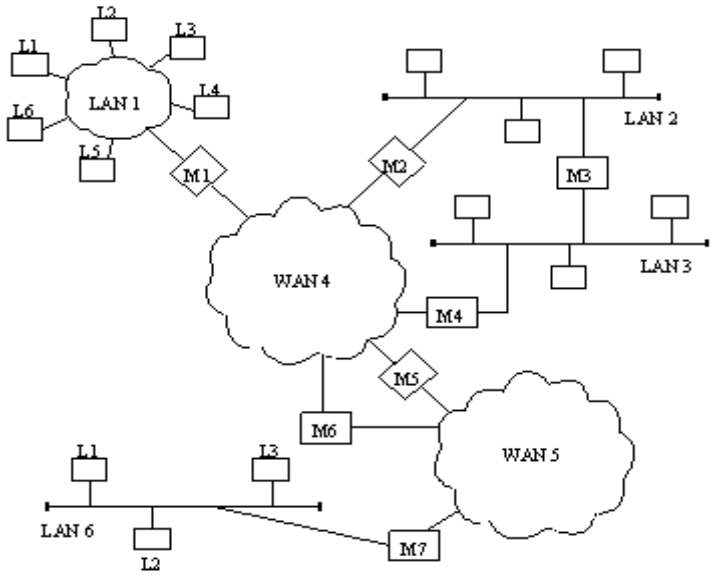


Рисунок 43 - Структура интерсети, построенной на основе маршрутизаторов

M1, M2, ... , M7 – маршрутизаторы;
LAN1, LAN2, LAN3, WAN4, WAN5, LAN6 - уникальные номера сетей в

едином формате;
L1, L2, ... - локальные номера узлов (дублируются, разный формат).

В отличие от моста и коммутатора, которые не знают, как связаны сегменты друг с другом за пределами их портов, маршрутизатор видит всю картину связей подсетей друг с другом, поэтому он может выбрать правильный маршрут и при наличии нескольких альтернативных маршрутов. Решение о выборе того или иного маршрута принимается каждым маршрутизатором, через который проходит сообщение.

Для того, чтобы составить карту связей в сети, маршрутизаторы обмениваются специальными служебными сообщениями, в которых содержится информация о тех связях между подсетями (эти подсети подключены к ним непосредственно или же они узнали эту информацию от других маршрутизаторов).

Построение графа связей между подсетями и выбор оптимального по какому-либо критерию маршрута на этом графе представляют собой сложную задачу. При этом могут использоваться разные критерии выбора маршрута - наименьшее количество промежуточных узлов, время, стоимость или надежность передачи данных.

Маршрутизаторы позволяют объединять сети с различными принципами организации в единую сеть, которая в этом случае часто называется *интерсеть* (internet). Название интерсеть подчеркивает ту особенность, что образованное с помощью маршрутизаторов объединение компьютеров представляет собой совокупность нескольких сетей, сохраняющих большую степень автономности, чем несколько логических сегментов одной сети.

В каждой из сетей, образующих интерсеть, сохраняются присущие им принципы адресации узлов и протоколы обмена информацией. Поэтому маршрутизаторы могут объединять не только локальные сети с различной технологией, но и локальные сети с глобальными.

Маршрутизаторы не только объединяют сети, но и надежно защищают их друг от друга. Причем эта изоляция осуществляется гораздо проще и надежнее, чем с помощью мостов и коммутаторов. Например, при поступлении кадра с неправильным адресом мост или коммутатор обязан повторить его на всех своих портах, что делает сеть незащищенной от некорректно работающего узла. Маршрутизатор же в таком случае просто отказывается передавать "неправильный" пакет дальше, изолируя дефектный узел от остальной сети.

Кроме того, маршрутизатор предоставляет администратору удобные средства фильтрации потока сообщений за счет того, что сам распознает многие поля служебной информации в пакете и позволяет их именовать понятным администратору образом.

Кроме фильтрации, маршрутизатор может обеспечивать приоритетный порядок обслуживания буферизованных пакетов, когда на основании некоторых признаков пакетам предоставляются преимущества при выборе из очереди.

Маршрутизаторы часто путают с мостами. Такое положение дел объясняется тем, что многие устройства сочетают в себе функции и мостов, и маршрутизаторов.

"Чистый" мост анализирует заголовки кадра канального уровня и не просматривает (а тем более не модифицирует) пакеты сетевого уровня внутри пакетов. Мост не знает и не должен знать, какие пакеты - IP, IPX - содержит в поле кадр, передаваемый из одной локальной сети в другую.

Маршрутизатор, наоборот, знает очень хорошо, с какими пакетами он работает, анализирует заголовки этих пакетов и принимает решение в соответствии с содержащейся там адресной информацией. С другой стороны, когда маршрутизатор передает пакет на канальный уровень, он не знает и не должен знать о том, в какой кадр данный пакет будет помещен - Ethernet, Token Ring или какой-либо иной.

Маршрутизация - процесс определения в сети пути, по которому вызов либо блок данных может достигнуть адресата. Основная же задача маршрутизации - переключение трафика. Переключение - это процесс приема сообщения, выбора подходящего маршрута дальнейшего следования и отправка его по этому маршруту. Данная операция обслуживается четырьмя различными процессами: входным драйвером, процессом выбора маршрута, очередью и выходным драйвером.

Маршрутизация выполняется на основе данных, содержащихся в таблице маршрутов. Строка в таблице маршрутов состоит из следующих полей:

– адрес сети назначения;

– адрес следующего маршрутизатора (то есть узла, который знает, куда дальше отправить дейтаграмму, адресованную в сеть назначения);

– вспомогательные поля.

Маршрутизация решает две задачи:

– выбор оптимального, по некоторому критерию, пути продвижения информации от источника к пункту назначения через объединенную сеть;

– транспортировка информационных блоков (пакетов) по выбранному маршруту, или коммутация.

3.4.1 Примеры маршрутизации

Рассмотрим процесс маршрутизации на примере. Допустим (рисунок 44), хосты A и B находятся в сети 1, сеть 1 соединяется с сетью 2 с помощью маршрутизатора G1. К сети 2 подключен маршрутизатор G2, соединяющий ее с сетью 3, в которой находится хост C.

Таблица маршрутов хоста A выглядит, например, так:

Сеть 1 – A

Прочие сети - G1

Это означает, что дейтаграммы, адресованные узлам сети 1, отправляет сам хост A (так как это его локальная сеть), а дейтаграммы, адресованные в

любую другую сеть (это называется маршрут по умолчанию), хост А отправляет маршрутизатору G1, чтобы тот занялся их дальнейшей судьбой.

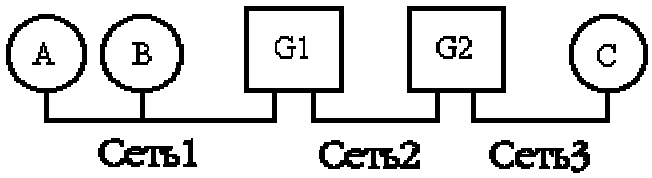


Рисунок 44 - Пример маршрутизации

Предположим, хост А посылает дейтаграмму хосту В. В этом случае, поскольку адрес В принадлежит той же сети, что и А, из таблицы маршрутов хоста А определяется, что доставка осуществляется непосредственно самим хостом А.

Если хост А отправляет дейтаграмму хосту С, то он определяет по IP-адресу С, что хост С не принадлежит к сети 1. Согласно таблице маршрутов А, все дейтаграммы с пунктами назначения, не принадлежащими сети 1, отправляются на маршрутизатор G1 (это называется маршрут по умолчанию). При этом хост А не знает, что маршрутизатор G1 будет делать с его дейтаграммой и каков будет ее дальнейший маршрут - это забота исключительно G1. G1 в свою очередь по своей таблице маршрутов определяет, что все дейтаграммы, адресованные в сеть 3, должны быть пересланы на маршрутизатор G2. Это может быть, как явно указано в таблице, находящейся на G1, в виде

Сеть 3 - G2,

так и указано в виде маршрута по умолчанию.

На этом функции G1 заканчиваются, дальнейший путь дейтаграммы ему неизвестен и его не интересует. Маршрутизатор G2, получив дейтаграмму, определяет, что она адресована в одну из сетей (№3), к которым он присоединен непосредственно, и доставляет дейтаграмму на хост С.

Пример подключения локальной сети организации к Интернет.

Рассмотрим реальный пример подключения к Интернет локальной сети организации (рисунок 45). IP-адрес локальной сети - 194.84.124.0/24 (сеть класса С). В эту сеть включен маршрутизатор G1. IP-интерфейс этого маршрутизатора, подключенный к локальной сети Ethernet, имеет адрес 194.84.124.1. Второй IP-интерфейс маршрутизатора подключен к выделенной линии (синхронный последовательный канал), ведущей к провайдеру Интернет. К другому концу этой линии подключен IP-интерфейс маршрутизатора G2, принадлежащего провайдеру. Эти два интерфейса образуют отдельную сеть 194.84.0.116/30. В этой сети на номер интерфейса отведено всего 2 бита — 4 варианта адресов, из которых один (00) обозначает саму сеть, один (11) — широковещательный; таким образом, в подобной сети может находиться всего 2 узла — это минимальная возможная сеть. Интерфейс маршрутизатора G1 в сети 194.84.0.116/30 имеет адрес 194.84.0.117, а маршрутизатора G2 —

194.84.0.118. Маршрутизатор G2 имеет еще некоторое количество интерфейсов, к части которых подключены выделенные линии от других клиентов, к части — локальные сети коммуникационного центра, другие маршрутизаторы и магистральные линии дальней связи.

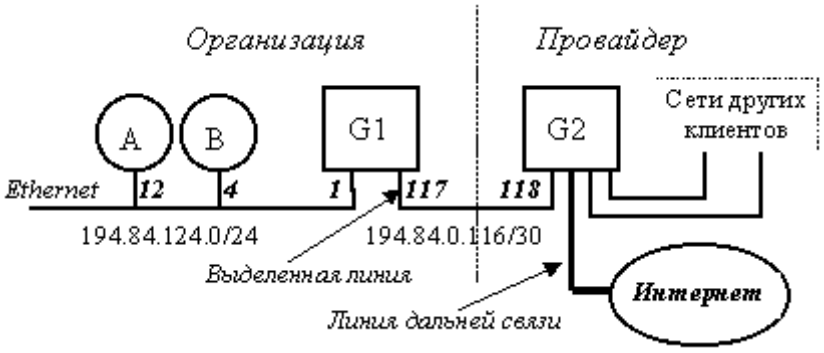


Рисунок 45 - Подключение локальной сети к Интернет

Таблицы маршрутов. Рассмотрим примеры маршрутных таблиц, с которыми имеет дело администратор локальной сети 194.84.124.0/24:

Таблица 6 - Маршруты рядового хоста с адресом 194.84.124.4 (хост B)

Destination	Gateway	Flags	Interface
127.0.0.1	127.0.0.1	UH	lo0
194.84.124.0	194.84.124.4	U	le0
0.0.0.0	194.84.124.1	UG	

Значения флагов: **U** (Up) - маршрут работает; **H** (Host) - пунктом назначения является отдельный узел (хост), а не сеть; **G** (Gateway) - маршрут к сети назначения проходит через один или несколько промежуточных маршрутизаторов. Интерфейс le0 обозначает Ethernet, lo0 - интерфейс обратной связи (loopback).

Значение первой записи очевидно, вторая запись определяет, что дейтаграммы, адресованные в локальную сеть, хост отправляет самостоятельно через свой интерфейс le0. Третья запись (маршрут по умолчанию) устанавливает, что все остальные дейтаграммы передаются на адрес 194.84.124.1, который является адресом следующего маршрутизатора (флаг G), для дальнейшей пересылки. Чтобы определить способ достижения самого маршрутизатора, следует, очевидно, обратиться ко второй строке таблицы, так как адрес маршрутизатора принадлежит сети 194.84.124.0.

Заметим, что в этой таблице для простоты опущены маски сетей.

Пример таблицы маршрутов маршрутизатора, соединяющего локальную сеть с провайдером Интернет по выделенному каналу (G1 на рисунке 45):

Таблица 7 – Маршруты маршрутизатора, соединяющего локальную сеть с провайдером Интернет по выделенному каналу

Destination	Mask	Gateway	Interface
194.84.124.0	255.255.255.0	194.84.124.1	le0
194.84.0.116	255.255.255.252	194.84.0.117	se0
0.0.0.0	0.0.0.0	194.84.0.118	

В таблице явно показаны маски сетей.

Первые две записи говорят о том, что маршрутизатор самостоятельно, через свои соответствующие IP-интерфейсы отправляет дейтаграммы, адресованные в сети, к которым он подключен непосредственно. Все остальные дейтаграммы перенаправляются к G2 (194.84.0.118). Интерфейс se0 обозначает последовательный (serial) канал - выделенную линию.

Таблица маршрутов может заполняться различными способами. Статическая маршрутизация применяется в том случае, когда используемые маршруты не могут измениться в течение времени, например, для выше обсужденных хоста и маршрутизатора, где просто отсутствуют какие-либо альтернативные маршруты. Статические маршруты конфигурируются администратором сети или конкретного узла.

В случае объединения сетей со сложной топологией, когда существует несколько вариантов маршрутов от одного узла к другому и (или) когда состояние сетей (топология, качество каналов связи) изменяется с течением времени, таблицы маршрутов составляются динамически с помощью различных протоколов маршрутизации. Протоколы маршрутизации на основании тех или иных алгоритмов динамически редактируют таблицу маршрутов, то есть вносят и удаляют записи, при этом часть записей может по-прежнему статически вноситься администратором.

В зависимости от алгоритма работы различают *дистанционно-векторные* протоколы (distance vector protocols) и протоколы *состояния связей* (link state protocols). По области применения существует разделение на протоколы *внешней* (exterior) и *внутренней* (interior) маршрутизации.

3.5 Контрольные вопросы

1) Повторители и концентраторы.
2) Мосты и коммутаторы.
3) Принцип работы коммутатора.
4) Классы коммутаторов Ethernet.
5) Маршрутизация.

3.6 Тесты

1) Какие устройства объединяют сети на физическом уровне?
а) повторители;
б) мосты;
в) коммутаторы;
г) среди перечисленных устройств нет правильных.

2) Какие устройства объединяют сети на канальном уровне?
а) мосты, коммутаторы;
б) мосты;
в) коммутаторы;
г) среди перечисленных устройств нет правильных.

3) Какие устройства образуют на отдельных физических отрезках кабеля среду передачи данных - логический сегмент?
а) концентраторы;
б) коммутаторы;
в) маршрутизаторы.

4) Какие типы коммутаторов обеспечивают соединение со скоростью передачи среды между парой незанятых сегментов Ethernet?
а) магистральные;
б) коммутаторы для рабочих групп;
в) и те и другие.

5) Из каких полей состоит таблица маршрутов?
а) адрес сети назначения, адрес следующего маршрутизатора;
б) адрес сети назначения, адрес следующего маршрутизатора, вспомогательные поля;
с) адрес сети назначения, адрес следующего маршрутизатора, адрес предыдущего маршрутизатора.

6) Ядром какого устройства является коммутационная матрица, обеспечивающая передачу данных между любыми двумя точками, или быстродействующая шина, через которую любой порт может передать информацию любому другому порту?
а) маршрутизатор;
б) коммутатор;
в) концентратор.

7) Для каких устройств сеть представляется набором MAC-адресов?
а) коммутаторы;
б) маршрутизаторы;
в) мосты, коммутаторы;

г) мосты;
д) концентраторы.

8) Таблица какого устройства представлена на рисунке?

а) маршрутизатор;
б) концентратор;
в) мост;
г) коммутатор.

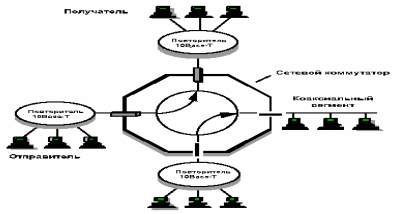
MAC-адрес	Номер порта
A	1
B	2
C	3
D	4

9) Какие функции выполняет «чистый» мост?

а) просматривает пакеты сетевого уровня;
б) модифицирует пакеты сетевого уровня;
в) а,б;
г) анализирует заголовки кадра канального уровня.

10) Какой тип коммутаторов представлен на рисунке:

а) магистральный;
б) для рабочих групп;
в) нет правильного.



11) Какие устройства объединяют сети в единую сеть – интерсеть?

а) маршрутизаторы;
б) концентраторы;
в) мосты;
г) коммутаторы.

12) В Ядре какого устройства образуется сверхбыстрая «виртуальная сеть»?

а) мост;
б) повторитель;
в) коммутатор;
г) маршрутизатор.

13) Что не входит в функции моста:

а) анализ поступающих фреймов;
б) принимают решения о пересылке фреймов к месту назначения;
в) все не правильно;
г) все правильно.

14) Как называется многопортовый повторитель?

а) концентратор;
б) hub;
в) коммутатор.

15) Какие задачи решает маршрутизация?

а) выбор оптимального по некоторому критерию пути продвижения информации от источника к пункту назначения через объединенную сеть;

б) транспортировка информационных блоков (пакетов) по выбранному маршруту, или коммутация;

в) все неправильно;

г) все правильно.

4 Базовые технологии локальных сетей

4.1 Технология Ethernet

Ethernet - это самый распространенный на сегодняшний день стандарт локальных сетей.

Когда говорят Ethernet, то под этим обычно понимают любой из вариантов этой технологии. В более узком смысле *Ethernet - это сетевой стандарт*, основанный на экспериментальной сети Ethernet Network, которую фирма Xerox разработала и реализовала в 1975 году. Метод доступа был опробован еще раньше: во второй половине 60-х годов в радиосети Гавайского университета использовались различные варианты случайного доступа к общей радиосреде, получившие общее название Aloha. В 1980 году фирмы DEC, Intel и Xerox совместно разработали и опубликовали стандарт Ethernet версии II для сети, построенной на основе коаксиального кабеля, который стал последней версией фирменного стандарта Ethernet. Поэтому фирменную версию стандарта Ethernet называют стандартом Ethernet DIX или Ethernet II /14/.

На основе стандарта *Ethernet DIX* был разработан стандарт *IEEE 802.3*, который во многом совпадает со своим предшественником, но некоторые различия все же имеются. В то время как в стандарте *IEEE 802.3* различаются уровни MAC и LLC, в оригинальном Ethernet оба эти уровня объединены в единый канальный уровень, В Ethernet DIX определяется протокол тестирования конфигурации (*Ethernet Configuration Test Protocol*), который отсутствует в *IEEE 802.3*. Несколько отличается и формат кадра, хотя минимальные и максимальные размеры кадров в этих стандартах совпадают. Часто для того, чтобы отличить Ethernet, определенный стандартом IEEE, и фирменный Ethernet DIX, первый называют технологией 802.3, а за фирменным оставляют название Ethernet без дополнительных обозначений.

В зависимости от типа физической среды стандарт IEEE 802.3 имеет различные модификации:

- 10Base-5;
- 10Base-2;
- 10Base-T;

– 10Base-FL;
– 10Base-FB.

В 1995 году был принят стандарт Fast Ethernet, который во многом не является самостоятельным стандартом, о чем говорит и тот факт, что его описание просто является дополнительным разделом к основному стандарту 802.3 - разделом 802.3ч. Аналогично, принятый в 1998 году стандарт Gigabit Ethernet описан в разделе 802.3z основного документа.

Для передачи двоичной информации по кабелю для всех вариантов физического уровня технологии Ethernet, обеспечивающих пропускную способность 10 Мбит/с, используется манчестерский код.

Все виды стандартов Ethernet (в том числе Fast Ethernet и Gigabit Ethernet) используют один и тот же метод разделения среды передачи данных - метод CSMA/CD.

4.1.1 Метод доступа CSMA/CD

В сетях Ethernet используется метод доступа к среде передачи данных, называемый *методом коллективного доступа с опознаванием несущей и обнаружением коллизий (carrier-sense-multiply-access with collision detection, CSMA/CD)*.

Этот метод используется исключительно в сетях с общей шиной (к которым относятся и радиосети, породившие этот метод). Все компьютеры такой сети имеют непосредственный доступ к общей шине, поэтому она может быть использована для передачи данных между любыми двумя узлами сети. Простота схемы подключения - это один из факторов, определивших успех стандарта Ethernet. Говорят, что кабель, к которому подключены все станции, работает в режиме *коллективного доступа (multiply-access,MA)*.

Все данные, передаваемые по сети, помещаются в кадры определенной структуры и снабжаются уникальным адресом станции назначения. Затем кадр передается по кабелю. Все станции, подключенные к кабелю, могут распознать факт передачи кадра, и та станция, которая узнает собственный адрес в заголовках кадра, записывает его содержимое в свой внутренний буфер, обрабатывает полученные данные и посылает по кабелю кадр-ответ. Адрес станции-источника также включен в исходный кадр, поэтому станция-получатель знает, кому нужно послать ответ.

При описанном подходе возможна ситуация, когда две станции одновременно пытаются передать кадр данных по общему кабелю (рисунок 54). Для уменьшения вероятности этой ситуации непосредственно перед отправкой кадра передающая станция слушает кабель (то есть принимает и анализирует возникающие на нем электрические сигналы), чтобы обнаружить, не передается ли уже по кабелю кадр данных от другой станции. Если *опознается несущая (carrier-sense, CS)*, то станция откладывает передачу своего кадра до окончания чужой передачи, и только потом пытается вновь его передать. Но даже при таком алгоритме две станции одновременно могут решить, что по шине в данный момент времени нет передачи, и начать одновременно передавать свои

кадры. Говорят, что при этом происходит *коллизия*, так как содержимое обоих кадров сталкивается на общем кабеле, что приводит к искажению информации.

Чтобы корректно обработать коллизию, все станции одновременно наблюдают за возникающими на кабеле сигналами. Если передаваемые и наблюдаемые сигналы отличаются, то фиксируется *обнаружение коллизии (collision detection, CD)*. Для увеличения вероятности немедленного обнаружения коллизии всеми станциями сети, ситуация коллизии усиливается посылкой в сеть станциями, начавшими передачу своих кадров, специальной последовательности битов, называемой *jam-последовательностью*.

После обнаружения коллизии передающая станция обязана прекратить передачу и ожидать в течение короткого случайного интервала времени, а затем может снова сделать попытку передачи кадра (рисунок 46).

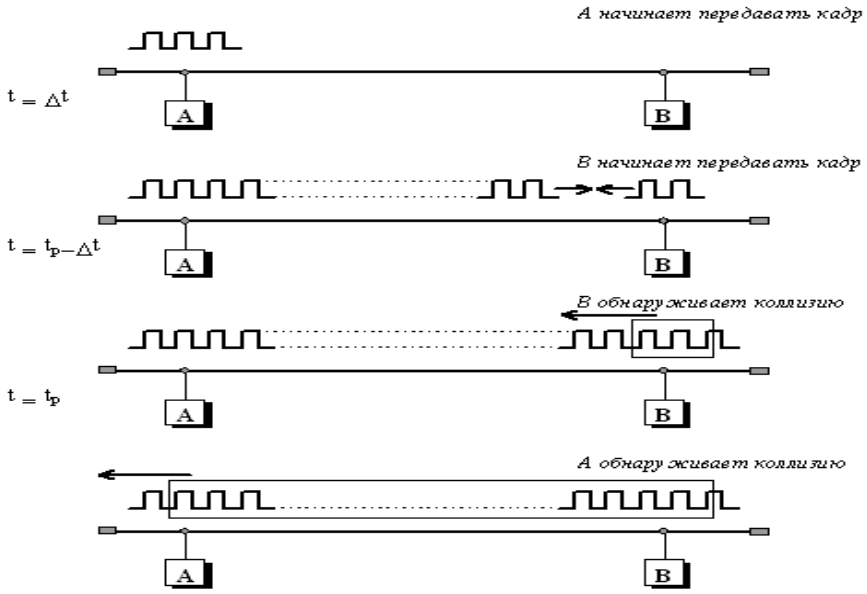


Рисунок 46 - Схема возникновения коллизии в методе случайного доступа CSMA/CD (tp - задержка распространения сигнала между станциями A и B)

Метод CSMA/CD определяет основные временные и логические соотношения, гарантирующие корректную работу всех станций в сети:

- Между двумя последовательно передаваемыми по общей шине кадрами информации должна выдерживаться пауза в 9.6 мкс; эта пауза нужна для приведения в исходное состояние сетевых адаптеров узлов, а также для предотвращения монопольного захвата среды передачи данных одной станцией.
- При обнаружении коллизии (условия ее обнаружения зависят от применяемой физической среды) станция выдает в среду специальную 32-х битную последовательность (jam-последовательность), усиливающую явление коллизии для более надежного распознавания ее всеми узлами сети.
- После обнаружения коллизии каждый узел, который передавал кадр и столкнулся с коллизией, после некоторой задержки пытается повторно передать

свой кадр. Узел делает максимально 16 попыток передачи этого кадра информации, после чего отказывается от его передачи. Величина задержки выбирается как равномерно распределенное случайное число из интервала, длина которого экспоненциально увеличивается с каждой попыткой. Такой алгоритм выбора величины задержки снижает вероятность коллизий и уменьшает интенсивность выдачи кадров в сеть при ее высокой загрузке.

Все параметры протокола Ethernet подобраны таким образом, чтобы при нормальной работе узлов сети коллизии всегда четко распознавались. Именно для этого минимальная длина поля данных кадра должна быть не менее 46 байт. Длина кабельной системы выбирается таким образом, чтобы за время передачи кадра минимальной длины сигнал коллизии успел бы распространиться до самого дальнего узла сети. Поэтому для скорости передачи данных 10 Мб/с, используемой в стандартах Ethernet, максимальное расстояние между двумя любыми узлами сети не должно превышать 2500 метров.

Независимо от реализации физической среды, все сети Ethernet должны удовлетворять двум ограничениям, связанным с методом доступа:

- максимальное расстояние между двумя любыми узлами не должно превышать 2500 м;
- в сети не должно быть более 1024 узлов.

4.1.2 Спецификации физической среды Ethernet

Исторически первые сети технологии Ethernet были созданы на коаксиальном кабеле диаметром 0,5 дюйма. В дальнейшем были определены и другие спецификации физического уровня для стандарта Ethernet, позволяющие использовать различные среды передачи данных. Метод доступа CSMA/CD и все временные параметры остаются одними и теми же для любой спецификации физической среды технологии Ethernet 10 Мбит/с.

Физические спецификации технологии Ethernet на сегодняшний день включают следующие среды передачи данных.

– **10Base-5** - коаксиальный кабель диаметром 0,5 дюйма, называемый «толстым» коаксиалом. Имеет волновое сопротивление 50 Ом. Максимальная длина сегмента - 500 метров (без повторителей).

– **10Base-2** - коаксиальный кабель диаметром 0,25 дюйма, называемый «тонким» коаксиалом. Имеет волновое сопротивление 50 Ом. Максимальная длина сегмента - 185 метров (без повторителей).

– **10Base-T** - кабель на основе неэкранированной витой пары. Образует звездообразную топологию на основе концентратора. Расстояние между концентратором и конечным узлом - не более 100 м.

– **10Base-F** - волоконно-оптический кабель. Топология аналогична топологии стандарта 10Base-T. Имеется несколько вариантов этой спецификации - FOIRL (расстояние до 1000 м), 10Base-FL (расстояние до 2000 м), 10Base-FB (расстояние до 2000 м).

Число 10 в указанных выше названиях обозначает битовую скорость передачи данных этих стандартов - 10 Мбит/с,

Слово Base - метод передачи на одной базовой частоте 10 МГц (в отличие от методов, использующих несколько несущих частот, которые называются Broadband - широкополосными). Последний символ в названии стандарта физического уровня обозначает тип кабеля.

4.1.3 Стандарт 10Base-5

Стандарт 10Base-5 в основном соответствует экспериментальной сети Ethernet фирмы Xerox и может считаться классическим Ethernet. Он использует в качестве среды передачи данных коаксиальный кабель с волновым сопротивлением 50 Ом, диаметром центрального медного провода 2,17 мм и внешним диаметром около 10 мм («толстый» Ethernet). Различные компоненты сети, выполненной на толстом коаксиале, показаны на рисунке 47.

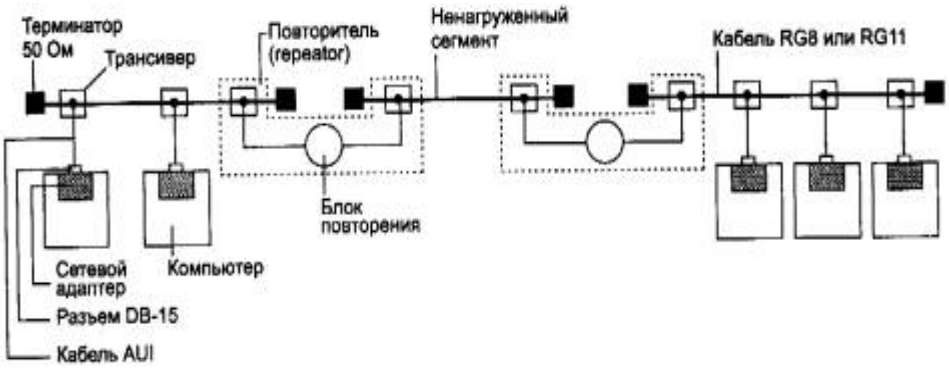


Рисунок 47 - Компоненты физического уровня сети стандарта 10 Base-5, состоящей из трех сегментов

Кабель используется как моноканал для всех станций. Сегмент кабеля имеет максимальную длину 500 м (без повторителей) и должен иметь на концах согласующие *терминаторы* сопротивлением 50 Ом, поглощающие распространяющиеся по кабелю сигналы и препятствующие возникновению отраженных сигналов. При отсутствии терминаторов («заглушек») в кабеле возникают стоячие волны, так что одни узлы получают мощные сигналы, а другие - настолько слабые, что их прием становится невозможным.

Станция должна подключаться к кабелю при помощи приемопередатчика - *трансивера (transmitter+Teceiver = transceiver).* Трансивер устанавливается непосредственно на кабеле и питается от сетевого адаптера компьютера. Трансивер может подсоединяться к кабелю как методом прокалывания, обеспечивающим непосредственный физический контакт, так и бесконтактным методом.

Трансивер соединяется с сетевым адаптером интерфейсным кабелем *AUI (Attachment Unit Interface)* длиной до 50 м, состоящим из 4 витых пар (адаптер должен иметь разъем AUI). Наличие стандартного интерфейса между трансивером и остальной частью сетевого адаптера очень полезно при переходе с одного типа кабеля на другой. Для этого достаточно только заменить

трансивер, а остальная часть сетевого адаптера остается неизменной, так как она отрабатывает протокол уровня MAC. При этом необходимо только, чтобы новый трансивер (например, трансивер для витой пары) поддерживал стандартный интерфейс AUI.

Допускается подключение к одному сегменту не более 100 трансиверов, причем расстояние между подключениями трансиверов не должно быть меньше 2,5 м. На кабеле имеется разметка через каждые 2,5 м, которая обозначает точки подключения трансиверов. При подсоединении компьютеров, в соответствии с разметкой влияние стоячих волн в кабеле на сетевые адаптеры сводится к минимуму.

Трансивер - это часть сетевого адаптера, которая выполняет следующие функции:

- прием и передача данных с кабеля на кабель;
- определение коллизий на кабеле;
- электрическая развязка между кабелем и остальной частью адаптера;
- защита кабеля от некорректной работы адаптера.

Последнюю функцию иногда называют *«контролем болтливости»,* что является буквальным переводом соответствующего английского термина *(jabber control).*

Стандарт 10Base-5 определяет возможность использования в сети повторителя. Повторитель принимает сигналы из одного сегмента кабеля и побитно синхронно повторяет их в другом сегменте, улучшая форму и мощность импульсов, а также синхронизируя импульсы. Повторитель состоит из двух (или нескольких) трансиверов, которые присоединяются к сегментам кабеля, а также блока повторения со своим тактовым генератором. Для лучшей синхроннизации передаваемых бит повторитель задерживает передачу нескольких первых бит преамбулы кадра, за счет чего увеличивается задержка передачи кадра с сегмента на сегмент.

Стандарт разрешает использование в сети не более 4 повторителей и, соответственно, не более 5 сегментов кабеля. При максимальной длине сегмента кабеля в 500 м это дает максимальную длину сети l0Base-5 в 2500 м. Только 3 сегмента из 5 могут быть нагруженными, то есть такими, к которым подключаются конечные узлы. Между нагруженными сегментами должны быть ненагруженные сегменты, так что максимальная конфигурация сети представляет собой два нагруженных крайних сегмента, которые соединяются ненагруженными сегментами еще с одним центральным нагруженным сегментом. На рисунке 47 был приведен пример сети Ethernet, состоящей из трех сегментов, объединенных двумя повторителями. Крайние сегменты являются нагруженными, а промежуточный - ненагруженным.

Правило применения повторителей в сети Ethernet 10Base-5 носит название *«правило 5-4-3у.* 5 сегментов, 4 повторителя, 3 нагруженных сегмента. Ограниченное число повторителей объясняется дополнительными задержками распространения сигнала, которые они вносят.

К *достоинствам* стандарта 10Base-5 относятся:

– хорошая защищенность кабеля от внешних воздействий;

– сравнительно большое расстояние между узлами;

– возможность простого перемещения рабочей станции в пределах длины кабеля AUI.

Недостатками 10Base-5 являются:

– высокая стоимость кабеля;

– сложность его прокладки из-за большой жесткости;

– потребность в специальном инструменте для заделки кабеля;

– остановка работы всей сети при повреждении кабеля или плохом соединении;

– необходимость заранее предусмотреть подводку кабеля ко всем возможным местам установки компьютеров.

4.1.4 Стандарт 10Base-2

Стандарт 10Base-2 использует в качестве передающей среды коаксиальный кабель с диаметром центрального медного провода 0,89 мм и внешним диаметром около 5 мм («тонкий» Ethernet). Кабель имеет волновое сопротивление 50 Ом.

Максимальная длина сегмента без повторителей составляет 185 м, сегмент должен иметь на концах согласующие терминаторы 50 Ом.

Станции подключаются к кабелю с помощью высокочастотного BNC Т-коннектора, который представляет собой тройник, один отвод которого соединяется с сетевым адаптером, а два других - с двумя концами разрыва кабеля. Максимальное количество станций, подключаемых к одному сегменту, - 30. Минимальное расстояние между станциями -1м. Кабель «тонкого» коаксиала имеет разметку для подключения узлов с шагом в 1 м.

Стандарт 10Base-2 также предусматривает использование повторителей, применение которых также должно соответствовать «правилу 5-4-3». В этом случае сеть будет иметь максимальную длину в 5x185 = 925 м. Очевидно, что это ограничение является более сильным, чем общее ограничение в 2500 метров.

Стандарт 10Base-2 очень близок к стандарту 10Base-5. Но трансиверы в нем объединены с сетевыми адаптерами за счет того, что более гибкий тонкий коаксиальный кабель может быть подведен непосредственно к выходному разъему платы сетевого адаптера, установленной в шасси компьютера. Кабель в данном случае «висит» на сетевом адаптере, что затрудняет физическое перемещение компьютеров.

Типичный состав сети стандарта 10Base-2, состоящей из одного сегмента кабеля, показан на (рисунок 48).

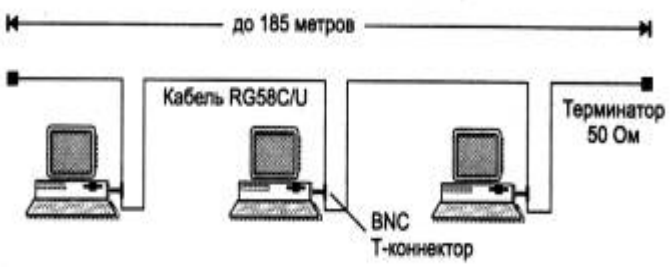


Рисунок 48 - Сеть стандарта 10Base-2

Реализация этого стандарта на практике приводит к наиболее простому решению для кабельной сети, так как для соединения компьютеров требуются только сетевые адаптеры, Т-коннекторы и терминаторы 50 Ом. Однако этот вид кабельных соединений наиболее сильно подвержен авариям и сбоям: кабель более восприимчив к помехам, чем «толстый» коаксиал.

Общим недостатком стандартов 10Base-5 и 10Base-2 является отсутствие оперативной информации о состоянии моноканала. Повреждение кабеля обнаруживается сразу же (сеть перестает работать), но для поиска отказавшего отрезка кабеля необходим специальный прибор - кабельный тестер.

4.1.5 Стандарт 10Base-T

Сети 10Base-T используют в качестве среды две *неэкранированные витые пары* (Unshielded Twisted Pair, UTP). Многопарный кабель на основе неэкранированной витой пары телефонные компании уже достаточно давно использовали для подключения телефонных аппаратов внутри зданий /15/.

Идея приспособить этот популярный вид кабеля для построения локальных сетей оказалась очень плодотворной, так как многие здания уже были оснащены нужной кабельной системой. Оставалось разработать способ подключения сетевых адаптеров и прочего коммуникационного оборудования к витой паре таким образом, чтобы изменения в сетевых адаптерах и программном обеспечении сетевых операционных систем были бы минимальными по сравнению с сетями Ethernet на коаксиале. Это удалось, поэтому переход на витую пару требует только замены трансивера сетевого адаптера или порта маршрутизатора, а метод доступа и все протоколы канального уровня остались теми же, что и в сетях Ethernet на коаксиале.

Конечные узлы соединяются по топологии «точка-точка» со специальным устройством – концентратором с помощью двух витых пар. Одна витая пара требуется для передачи данных от станции к повторителю (выход T_x сетевого адаптера), а другая - для передачи данных от повторителя к станции (вход R_x сетевого адаптера). На рисунке 49 показан пример трехпортового повторителя. Повторитель принимает сигналы от одного из конечных узлов и синхронно передает их на все свои остальные порты, кроме того, с которого поступили сигналы.

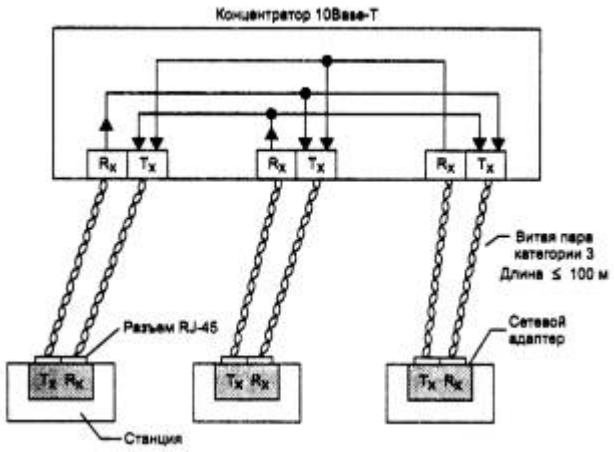


Рисунок 49 - Сеть стандарта 10Base-T: T_x - передатчик; R_x - приемник

Концентраторы 10Base-T можно соединять друг с другом с помощью тех же портов, которые предназначены для подключения конечных узлов. При этом нужно позаботиться о том, чтобы передатчик и приемник одного порта были соединены соответственно с приемником и передатчиком другого порта.

Для обеспечения синхронизации станций при реализации процедур доступа CSMA/CD и надежного распознавания станциями коллизий в стандарте определено максимально число концентраторов между любыми двумя станциями сети, а именно 4. Это правило носит название «правила 4-х хабов» и оно заменяет «правило 5-4-3», применяемое к коаксиальным сетям. При создании сети 10Base-T с большим числом станций концентраторы можно соединять друг с другом иерархическим способом, образуя древовидную структуру (рисунок 50).

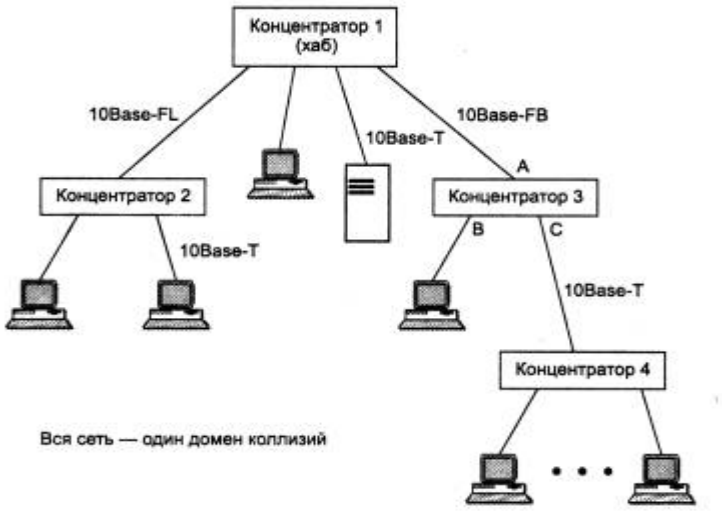


Рисунок 50 - Иерархическое соединение концентраторов Ethernet

Общее количество станций в сети 10Base-T не должно превышать общего предела в 1024, и для данного типа физического уровня это количество действительно можно достичь. Для этого достаточно создать двухуровневую иерархию концентраторов, расположив на нижнем уровне достаточное количество концентраторов с общим количеством портов 1024 (рисунок 51).

Рисунок 51 - Схема с максимальным количеством станций

Конечные узлы нужно подключить к портам концентраторов нижнего уровня. Правило 4-х хабов при этом выполняется - между любыми конечными узлами будет ровно 3 концентратора.

Максимальная длина сети в 2500 м здесь понимается как максимальное расстояние между любыми двумя конечными узлами сети (часто применяется также термин «максимальный диаметр сети»). Очевидно, что если между любыми двумя узлами сети не должно быть больше 4-х повторителей, то максимальный диаметр сети 10Base-T составляет 5*100 = 500 м.

Сети, построенные на основе стандарта 10Base-T, обладают по сравнению с коаксиальными вариантами Ethernet многими преимуществами. Эти преимущества связаны с разделением общего физического кабеля на отдельные кабельные отрезки, подключенные к центральному коммуникационному устройству.

4.1.6 Оптоволоконный Ethernet

В качестве среды передачи данных 10 мегабитный Ethernet использует оптическое волокно. Оптоволоконные стандарты в качестве основного типа кабеля рекомендуют достаточно дешевое оптическое волокно, обладающее полосой пропускания 500-800 МГц при длине кабеля 1 км.

Функционально сеть Ethernet на оптическом кабеле состоит из тех же элементов, что и сеть стандарта 10Base-T - сетевых адаптеров, многопортового повторителя и отрезков кабеля, соединяющих адаптер с портом повторителя. Как и в случае витой пары, для соединения адаптера с повторителем *используются два* оптоволокна - одно соединяет выход T_x адаптера со входом R_x повторителя, а другое - вход R_x адаптера с выходом T_x повторителя.

Стандарт FOIRL (Fiber Optic Inter-Repeater Link) представляет собой первый стандарт комитета 802.3 для использования оптоволокна в сетях Ethernet. Он гарантирует длину оптоволоконной связи между повторителями до 1 км при общей длине сети не более 2500 м. Максимальное число повторителей между любыми узлами сети - 4. Максимального диаметра в 2500 м здесь достичь можно, хотя максимальные отрезки кабеля между всеми 4 повторителями, а также между повторителями и конечными узлами недопустимы - иначе получится сеть длиной 5000 м.

Стандарт 10Base-FL представляет собой незначительное улучшение стандарта FOIRL. Увеличена мощность передатчиков, поэтому максимальное расстояние между узлом и концентратором увеличилось до 2000 м. Максимальное число повторителей между узлами осталось равным 4, а максимальная длина сети - 2500 м.

Стандарт 10Base-FB предназначен только для соединения повторителей. Конечные узлы не могут использовать этот стандарт для присоединения к портам концентратора. Между узлами сети можно установить до 5 повторителей 10Base-FB при максимальной длине одного сегмента 2000 м и максимальной длине сети 2740 м.

Как и в стандарте 10Base-T, оптоволоконные стандарты Ethernet разрешают соединять концентраторы только в древовидные иерархические структуры. Любые петли между портами концентраторов не допускаются.

4.1.7 Домен коллизий

В технологии Ethernet, независимо от применяемого стандарта физического уровня, существует понятие домена коллизий.

Домен коллизий (collision domain) - это часть сети Ethernet, все узлы которой распознают коллизию независимо от того, в какой части этой сети коллизия возникла. Сеть Ethernet, построенная на повторителях, всегда образует один домен коллизий. Домен коллизий соответствует одной разделяемой среде. Мосты, коммутаторы и маршрутизаторы делят сеть Ethernet на несколько доменов коллизий.

Приведенная на рисунке 50 сеть представляет собой один домен коллизий. Если, например, столкновение кадров произошло в концентраторе 4, то в соответствии с логикой работы концентраторов10Base-T сигнал коллизии распространится по всем портам всех концентраторов.

Если же вместо концентратора 3 поставить в сеть мост, то его порт С, связанный с концентратором 4, воспримет сигнал коллизии, но не передаст его на свои остальные порты, так как это не входит в его обязанности. Мост просто отработает ситуацию коллизии средствами порта С, который подключен к общей среде, где эта коллизия возникла. Если коллизия возникла из-за того, что мост пытался передать через порт С кадр в концентратор 4, то, зафиксировав сигнал коллизии, порт С приостановит передачу кадра и попытается передать его повторно через случайный интервал времени. Если порт С принимал в момент возникновения коллизии кадр, то он просто отбросит полученное начало кадра и будет ожидать, когда узел, передававший кадр через концентратор 4, не сделает повторную попытку передачи. После успешного принятия данного кадра в свой буфер мост передаст его на другой порт в соответствии с таблицей продвижения, например на порт А. Все события, связанные с обработкой коллизий портом С, для остальных сегментов сети, которые подключены к другим портам моста, останутся просто неизвестными.

Узлы, образующие один домен коллизий, работают синхронно, как единая распределенная электронная схема.

4.2 Технология Token Ring

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

помощью кадра специального формата, называемого *маркером* или *токеном (token)*.

Технология Token Ring был разработана компанией IBM в 1984 году, а затем передана в качестве проекта стандарта в комитет IEEE 802, который на ее основе принял в 1985 году стандарт 802.5.

Сети Token Ring работают с двумя битовыми скоростями - 4 и 16 Мбит/с. Смешение станций, работающих на различных скоростях, в одном кольце не допускается. Сети Token Ring, работающие со скоростью 16 Мбит/с, имеют некоторые усовершенствования в алгоритме доступа по сравнению со стандартом 4 Мбит/с.

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

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

4.2.1 Маркерный метод доступа к разделяемой среде

В сетях с *маркерным методом доступа* (а к ним, кроме сетей Token Ring, относятся сети FDDI) право на доступ к среде передается циклически от станции к станции по логическому кольцу.

В сети Token Ring кольцо образуется отрезками кабеля, соединяющими соседние станции. Таким образом, каждая станция связана со своей предшествующей и последующей станцией и может непосредственно обмениваться данными только с ними. Для обеспечения доступа станций к физической среде по кольцу циркулирует кадр специального формата и назначения - маркер. В сети Token Ring любая станция всегда непосредственно получает данные только от одной станции - той, которая является предыдущей в кольце. Такая станция называется *ближайшим активным соседом, расположенным выше по потоку* (данных) - *Nearest Active Upstream Neighbor, NAUN*. Передачу же данных станция всегда осуществляет своему ближайшему соседу вниз по потоку данных.

Получив маркер, станция анализирует его и при отсутствии у нее данных для передачи обеспечивает его продвижение к следующей станции. Станция, которая имеет данные для передачи, при получении маркера изымает его из кольца, что дает ей право доступа к физической среде и передачи своих данных. Затем эта станция выдает в кольцо кадр данных установленного формата последовательно по битам. Переданные данные проходят по кольцу всегда в одном направлении от одной станции к другой. Кадр снабжен адресом назначения и адресом источника.

Все станции кольца ретранслируют кадр побитно, как повторители. Если кадр проходит через станцию назначения, то, распознав свой адрес, эта станция копирует кадр в свой внутренний буфер и вставляет в кадр признак подтверждения приема. Станция, выдавшая кадр данных в кольцо, при обратном его получении с подтверждением приема изымает этот кадр из кольца и передает в сеть новый маркер для обеспечения возможности другим станциям сети передавать данные. Такой алгоритм доступа применяется в сетях Token Ring со скоростью работы 4 Мбит/с, описанных в стандарте 802.5.

На рисунке 52 описанный алгоритм доступа к среде иллюстрируется временной диаграммой.

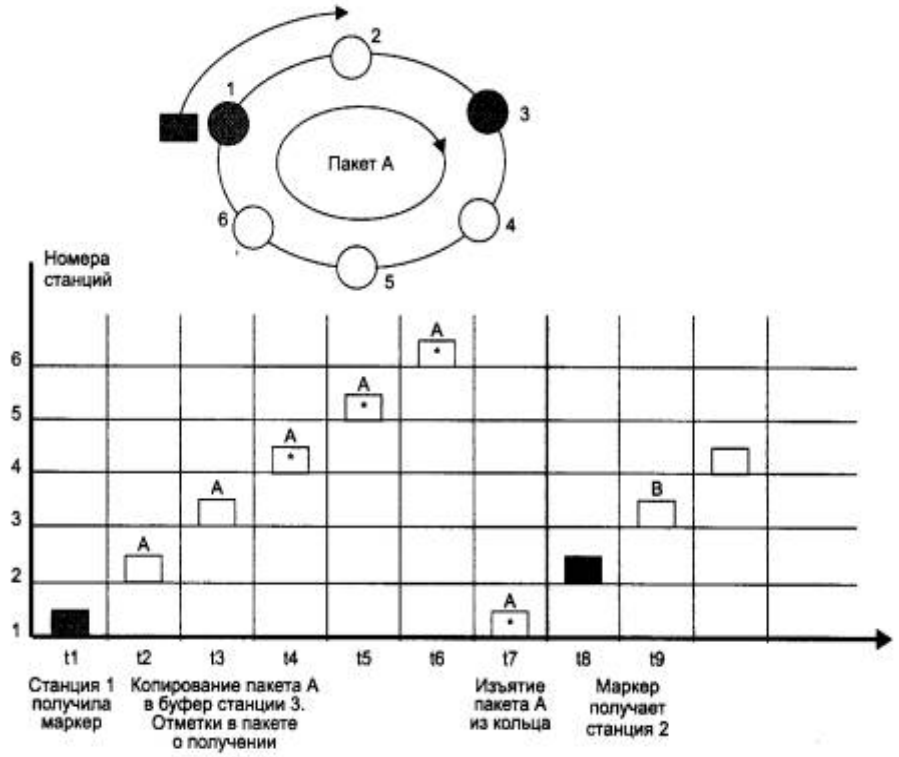


Рисунок 52 - Принцип маркерного доступа

Здесь показана передача пакета А в кольце, состоящем из 6 станций, от станции *1* к станции *3*. После прохождения станции назначения *3* в пакете А устанавливаются два признака - признак распознавания адреса и признак копирования пакета в буфер (что на рисунке отмечено звездочкой внутри пакета). После возвращения пакета в станцию *1* отправитель распознает свой пакет по адресу источника и удаляет пакет из кольца. Установленные станцией

3 признаки говорят станции-отправителю о том, что пакет дошел до адресата и был успешно скопирован им в свой буфер.

Время владения разделяемой средой в сети Token Ring ограничивается *временем удержания маркера (token holding time)*, после истечения которого станция обязана прекратить передачу собственных данных (текущий кадр разрешается завершить) и передать маркер далее по кольцу. Станция может успеть передать за время удержания маркера один или несколько кадров в зависимости от размера кадров и величины времени удержания маркера. Обычно время удержания маркера по умолчанию равно 10 мс.

В сетях Token Ring 16 Мбит/с используется также несколько другой алгоритм доступа к кольцу, называемый алгоритмом *раннего освобождения маркера (Early Token Release)*. В соответствии с ним станция передает маркер доступа следующей станции сразу же после окончания передачи последнего бита кадра, не дожидаясь возвращения по кольцу этого кадра с битом подтверждения приема. В этом случае пропускная способность кольца используется более эффективно, так как по кольцу одновременно продвигаются кадры нескольких станций. Тем не менее, свои кадры в каждый момент времени может генерировать только одна станция - та, которая в данный момент владеет маркером доступа. Остальные станции в это время только повторяют чужие кадры, так что принцип разделения кольца во времени сохраняется, ускоряется только процедура передачи владения кольцом.

Для различных видов сообщений, передаваемым кадрам, могут назначаться различные *приоритеты*: от 0 (низший) до 7 (высший). Решение о приоритете конкретного кадра принимает передающая станция (протокол Token Ring получает этот параметр через межуровневые интерфейсы от протоколов верхнего уровня, например прикладного). Маркер также всегда имеет некоторый уровень текущего приоритета. Станция имеет право захватить переданный ей маркер только в том случае, если приоритет кадра, который она хочет передать, выше (или равен) приоритета маркера. В противном случае станция обязана передать маркер следующей по кольцу станции.

За наличие в сети маркера, причем единственной его копии, отвечает активный монитор. Если активный монитор не получает маркер в течение длительного времени (например, 2,6 с), то он порождает новый маркер.

4.2.2 Физический уровень технологии Token Ring

Стандарт Token Ring фирмы IBM изначально предусматривал построение связей в сети с помощью концентраторов, называемых MAU (Multistation Access Unit) или MSAU (Multi-Station Access Unit), то есть устройствами многостанционного доступа (рисунок 53). Сеть Token Ring может включать до 260 узлов.

Концентратор Token Ring может быть активным или пассивным. Пассивный концентратор просто соединяет порты внутренними связями так, чтобы станции, подключаемые к этим портам, образовали кольцо. Такое устройство можно считать простым кроссовым блоком за одним исключением -

MSAU обеспечивает обход какого-либо порта, когда присоединенный к этому порту компьютер выключают. Такая функция необходима для обеспечения связности кольца вне зависимости от состояния подключенных компьютеров.

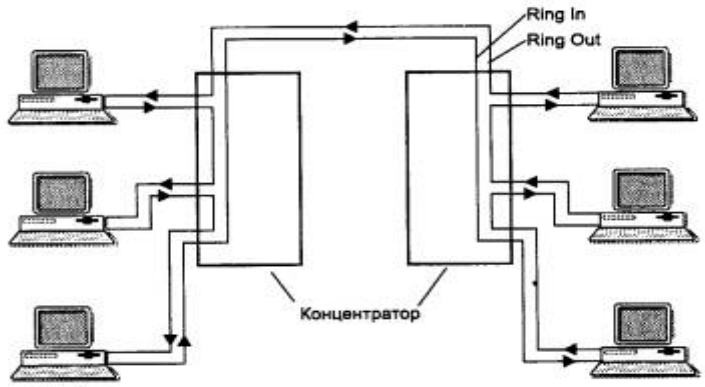


Рисунок 53 - Физическая конфигурация сети Token Ring

Активный концентратор выполняет функции регенерации сигналов и поэтому иногда называется повторителем, как в стандарте Ethernet.

Возникает вопрос - если концентратор является пассивным устройством, то каким образом обеспечивается качественная передача сигналов на большие расстояния, которые возникают при включении в сеть нескольких сот компьютеров? Ответ состоит в том, что роль усилителя сигналов в этом случае берет на себя каждый сетевой адаптер, а роль ресинхронизирующего блока выполняет сетевой адаптер активного монитора кольца. Каждый сетевой адаптер Token Ring имеет блок повторения, который умеет регенерировать и ресинхронизировать сигналы, однако последнюю функцию выполняет в кольце только блок повторения активного монитора.

Максимальная длина кольца Token Ring составляет 4000 м. Ограничения на максимальную длину кольца и количество станций в кольце в технологии Token Ring не являются такими жесткими, как в технологии Ethernet. Здесь эти ограничения во многом связаны со временем оборота маркера по кольцу (но не только - есть и другие соображения, диктующие выбор ограничений). Так, если кольцо состоит из 260 станций, то при времени удержания маркера в 10 мс маркер вернется в активный монитор в худшем случае через 2,6 с, а это время как раз составляет тайм-аут контроля оборота маркера.

Существует большое количество аппаратуры для сетей Token Ring, которая улучшает некоторые стандартные характеристики этих сетей: максимальную длину сети, расстояние между концентраторами, надежность (путем использования двойных колец).

4.3 Технология FDDI

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

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

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

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

В случае какого-либо вида отказа, когда часть первичного кольца не может передавать данные (например, обрыв кабеля или отказ узла), первичное кольцо объединяется со вторичным (рисунок 54), вновь образуя единое кольцо. Этот режим работы сети называется *Wrap,* то есть «свертывание» или «сворачивание» колец. Операция свертывания производится средствами концентраторов и/или сетевых адаптеров FDDI. Для упрощения этой процедуры данные по первичному кольцу всегда передаются в одном направлении (на диаграммах это направление изображается против часовой стрелки), а по вторичному - в обратном (изображается по часовой стрелке). Поэтому при образовании общего кольца из двух колец передатчики станций по-прежнему остаются подключенными к приемникам соседних станций, что позволяет правильно передавать и принимать информацию соседними станциями.

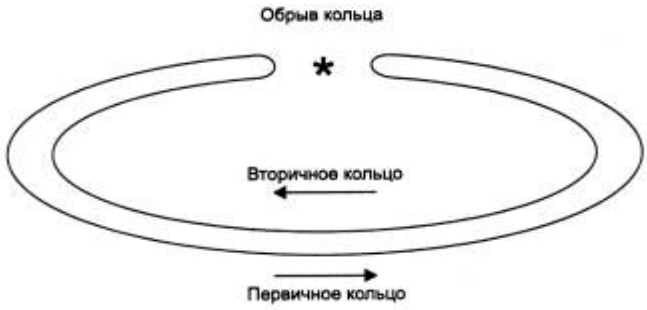


Рисунок 54 - Реконфигурация колец FDDI при отказе

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

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

Отличия метода доступа заключаются в том, что время удержания маркера в сети FDDI не является постоянной величиной, как в сети Token Ring. Это время зависит от загрузки кольца - при небольшой загрузке оно увеличивается, а при больших перегрузках может уменьшаться до нуля. Эти

изменения в методе доступа касаются только асинхронного трафика, который не критичен к небольшим задержкам передачи кадров. Для синхронного трафика время удержания маркера по-прежнему остается фиксированной величиной. Механизм приоритетов кадров, аналогичный принятому в технологии Token Ring, в технологии FDDI отсутствует. Разработчики технологии решили, что деление трафика на 8 уровней приоритетов избыточно и достаточно разделить трафик на два класса - асинхронный и синхронный, последний из которых обслуживается всегда, даже при перегрузках кольца.

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

Адреса уровня MAC имеют стандартный для технологий IEEE 802 формат. Формат кадра FDDI близок к формату кадра Token Ring, основные отличия заключаются в отсутствии полей приоритетов. Признаки распознавания адреса, копирования кадра и ошибки позволяют сохранить имеющиеся в сетях Token Ring процедуры обработки кадров станцией-отправителем, промежуточными станциями и станцией-получателем.

На рисунке 66 приведено соответствие структуры протоколов технологии FDDI семиуровневой модели OSI. FDDI определяет протокол физического уровня и протокол подуровня доступа к среде (MAC) канального уровня.

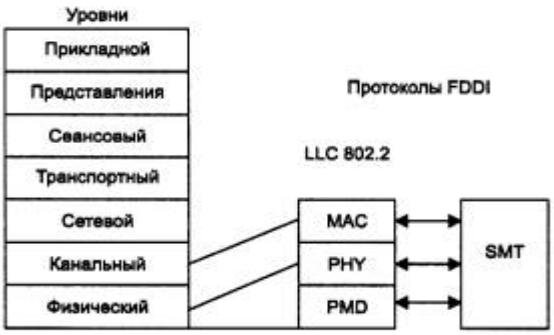


Рисунок 55 - Структура протоколов технологии FDDI

Как и во многих других технологиях локальных сетей, в технологии FDDI используется протокол подуровня управления каналом данных LLC, определенный в стандарте IEEE 802.2. Таким образом, несмотря на то, что технология FDDI была разработана и стандартизована институтом ANSI, а не комитетом IEEE, она полностью вписывается в структуру стандартов 802.

Отличительной особенностью технологии FDDI является уровень управления станцией - *Station Management (SMT)*. Именно уровень SMT выполняет все функции по управлению и мониторингу всех остальных уровней стека протоколов FDDI. В управлении кольцом принимает участие каждый узел сети FDDI. Поэтому все узлы обмениваются специальными кадрами SMT для управления сетью.

Отказоустойчивость сетей FDDI обеспечивается протоколами и других уровней: с помощью физического уровня устраняются отказы сети по физическим причинам, например из-за обрыва кабеля, а с помощью уровня MAC - логические отказы сети, например потеря нужного внутреннего пути передачи маркера и кадров данных между портами концентратора.

4.3.1 Особенности метода доступа FDDI

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

Если же станции кольца FDDI нужно передать асинхронный кадр (тип кадра определяется протоколами верхних уровней), то для выяснения возможности захвата маркера при его очередном поступлении станция должна измерить интервал времени, который прошел с момента предыдущего прихода маркера. Этот интервал называется временем оборота маркера (Token Rotation Time, TRT). Интервал TRT сравнивается с другой величиной - максимально допустимым временем оборота маркера по кольцу T_0рг. Если в технологии Token Ring максимально допустимое время оборота маркера является фиксированной величиной (2,6 с из расчета 260 станций в кольце), то в технологии FDDI станции договариваются о величине T_0рг во время инициализации кольца. Каждая станция может предложить свое значение T_0рг, в результате для кольца устанавливается минимальное из предложенных станциями времен. Это позволяет учитывать потребности приложений, работающих на станциях. Обычно синхронным приложениям (приложениям реального времени) нужно чаще передавать данные в сеть небольшими порциями, а асинхронным приложениям лучше получать доступ к сети реже, но большими порциями. Предпочтение отдается станциям, передающим синхронный трафик.

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

В стандарте FDDI предусмотрено наличие в сети конечных узлов - станций (Station), а также концентраторов (Concentrator). Для станций и концентраторов допустим любой вид подключения к сети - как одиночный, так и двойной. Соответственно такие устройства имеют соответствующие названия: SAS (Single Attachment Station), DAS (Dual Attachment Station), SAC (Single Attachment Concentrator) и DAC (Dual Attachment Concentrator).

Обычно концентраторы имеют двойное подключение, а станции - одинарное, как это показано на рисунке 56, хотя это и не обязательно. Чтобы

устройства легче было правильно присоединять к сети, их разъемы маркируются.

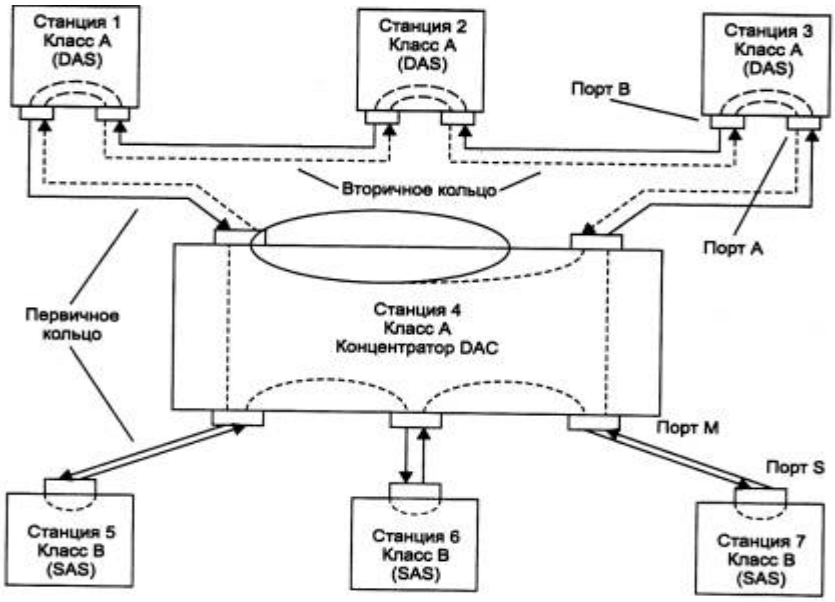


Рисунок 56 – Подключение узлов к кольцам FDDI

Разъемы типа А и В должны быть у устройств с двойным подключением, разъем М (Master) имеется у концентратора для одиночного подключения станции, у которой ответный разъем должен иметь тип S (Slave).

4.3.2 Сравнение FDDI с технологиями Ethernet и Token Ring

В таблице 13 представлены результаты сравнения технологии FDDI с технологиями Ethernet и Token Ring.

Технология FDDI разрабатывалась для применения в ответственных участках сетей - на магистральных соединениях между крупными сетями, например сетями зданий, а также для подключения к сети высокопроизводительных серверов. Поэтому главным для разработчиков было обеспечить высокую скорость передачи данных, отказоустойчивость на уровне протокола и большие расстояния между узлами сети. Все эти цели были достигнуты. В результате технология FDDI получилась качественной, но весьма дорогой. Даже появление более дешевого варианта для витой пары не намного снизило стоимость подключения одного узла к сети FDDI. Поэтому практика показала, что основной областью применения технологии FDDI стали магистрали сетей, состоящих из нескольких зданий, а также сети масштаба крупного города, то есть класса MAN. Для подключения клиентских компьютеров и даже небольших серверов технология оказалась слишком дорогой. А поскольку оборудование FDDI выпускается уже около 10 лет, значительного снижения его стоимости ожидать не приходится /16/.

В результате сетевые специалисты с начала 90-х годов стали искать пути создания сравнительно недорогих и в то же время высокоскоростных

технологий, которые бы так же успешно работали на всех этажах корпоративной сети, как это делали в 80-е годы технологии Ethernet и Token Ring. В таблице 8 представлено сопоставление технологии FDD с технологиями Ethernet, Token Ring

Таблица 8 – Характеристики технологий FDD, Ethernet, Token Ring

Характеристика	FDDI	Ethernet	Token Ring
Битовая скорость	100Мбит/с	10Мбит/с	16Мбит/с
Топология	Двойное кольцо деревьев	Шина/звезда	Звезда/кольцо
Метод доступа	Доля от времени оборота маркера	CSDMA/CD	Приоритетная система резервирования
Среда передачи данных	Оптоволокно, неэкранированная витая пара категории 5	Толстый коаксиал, тонкий коаксиал, витая пара категории 3, оптоволокно	Экранированная и неэкранированная витая пара, оптоволокно
Максимальная длина сети (без мостов)	200 км (100км на кольцо)	2500м	4000м
Максимальное расстояние между узлами	2км	2500м	100м
Максимальное количество узлов	500	1024	260 для экранированной витой пары, 72 для неакранированной витой пары

4.4 Контрольные вопросы

1) Что подразумевается под термином “Ethernet”.
2) Метод доступа CSMA/CD.
3) Технология Token Ring
4) Технология FDD
5) Сравнение технологий FDD, Ethernet и Token Ring

4.5 Тесты

1) С какой скоростью передавались данные в DIX-Ethernet?
   а) 10000 Кбит/сек;
   б) 1024 Кбит/сек;
   в) 10 Кбит/сек.

2) Какую топологию имели сети DIX-Ethernet?
   а) общая шина;
   б) звезда;
   в) кольцо.

3) Какое максимальное число последовательных коллизий может возникнуть в сети Ethernet?
   а) 2;
   б) 8;
   в) 4.

4) Какую длину имеет MAC-адрес?
   а) 32 бита;
   б) 48 бит;
   в) 32 байта.

5) Чему равна минимальная длина кадра Ethernet?
   а) 1024 байта;
   б) 64 байта;
   в) 1500 байт;
   г) 46 байт.

6) Чему равна максимальная длина кадра Ethernet?
   а) 1500 байт;
   б) 1024 байта;
   в) 1024 бита;
   г) 1.5 КБ.

7) Какой комитет занимается развитием спецификаций CSMA/CD?
   а) IEEE 802.5;
   б) IEEE 802.1;
   в) IEEE 802.3;
   г) IEEE 802.2.

8) Какая спецификация соответствует Ethernet по версии DIX?
   а) 10Base2;
   б) 10Base5;
   в) 10BaseT.

9) Чему равен размер максимального сегмента в сети Ethernet по версии DIX?

а) 100 метров;
б) 500 метров;
в) 1024 метра;
г) 185 метров.

10) Чему равен минимальный размер поля данных сети Ethernet?

а) 1500 байт;
б) 64 байта;
в) 46 байт.

11) Какой тип среды передачи данных используется в технологии 10Base5?

а) толстый коаксиальный кабель;
б) тонкий коаксиальный кабель;
в) волоконно-оптический кабель;
г) витая пара.

12) Какой тип среды передачи данных используется в технологии 10Base2?

а) толстый коаксиальный кабель;
б) волоконно-оптический кабель;
в) тонкий коаксиальный кабель;
г) витая пара.

13) Какой тип среды передачи данных используется в технологии 10BaseT?

а) толстый коаксиальный кабель;
б) волоконно-оптический кабель;
в) тонкий коаксиальный кабель;
г) витая пара.

14) Чему равен размер максимального сегмента в сети 10Base2?

а) 100 метров;
б) 500 метров;
в) 200 метров;
г) 185 метров.

15) Чему равен размер максимального сегмента в сети 10BaseT?

а) 100 метров;
б) 500 метров;
в) 185 метров;
г) 200 метров.

5 Основы TCP/IP

5.1 Классификация протоколов

Протокол - это набор правил и технических процедур, регулирующих порядок выполнения некоторой связи между компьютерами в компьютерной сети. Протоколы работают на разных уровнях модели OSI. Каждый протокол имеет определенное назначение, решает конкретные задачи и характеризуется такими показателями, как сложность, быстродействие, качество решения и надежность.Один протокол может решать задачи нескольких смежных уровней модели OSI /17/.

Все протоколы можно разделить на три группы, в зависимости от услуг, которые они предоставляют смежным уровням или прикладным процессам семиуровневой модели OSI (рисунок 57):

- прикладные услуги,
- транспортные услуги,
- сетевые услуги.

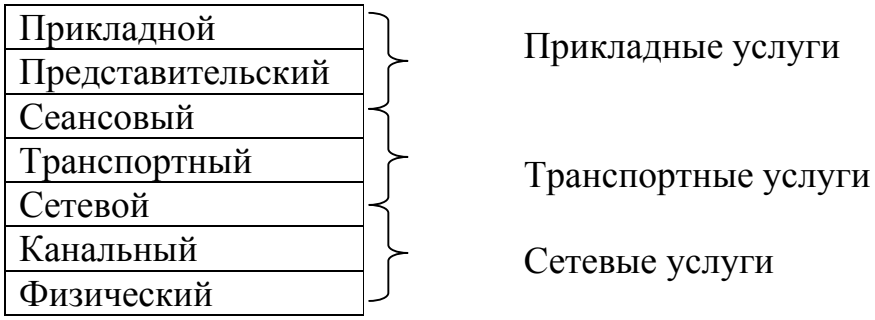


Рисунок 57 – Уровни модели OSI и типы протоков

Протоколы, обеспечивающие прикладные услуги, соответствуют трем верхним уровням модели OSI. Они контролируют взаимодействие приложений в сетевой среде.

К ним относятся такие протоколы как FTAM (File Transfer Access and Management), X.400, X.500, SMTP (Simple Mail Transfer Protocol), FTP (File Transfer Protocol), SNMP (Simple Network Management Protocol).

Протоколы, обеспечивающие транспортные услуги, поддерживают сеансы связи между компьютерами и надежный обмен данных между ними. К ним относятся протокол TCP (Transmission Control Protocol) и SPX (Sequential Packet Exchange).

Сетевые протоколы обеспечивают услуги связи. Для этого они управляют данными различного типа, связанными с адресацией и маршрутизацией пакетов, контролем ошибок и запросами на повторную передачу. Наиболее популярны протокол IP (Internet Protocol) и IPX (Internet Packet Exchange).

5.2 Сетевые протоколы

5.2.1 Протокол IP

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

Название данного протокола - *Intrenet Protocol* - отражает его суть: он должен передавать пакеты между сетями. В каждой очередной сети, лежащей на пути перемещения пакета, протокол IP вызывает средства транспортировки, принятые в этой сети, чтобы с их помощью передать этот пакет на маршрутизатор, ведущий к следующей сети, или непосредственно на узел-получатель /18/.

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

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

Имеется прямая связь между функциональной сложностью протокола и сложностью заголовка пакетов, которые этот протокол использует. Это объясняется тем, что основные служебные данные, на основании которых протокол выполняет то или иное действие, переносятся между двумя модулями, реализующими этот протокол на разных машинах, именно в полях заголовков пакетов. Поэтому очень полезно изучить назначение каждого поля заголовка IP-пакета, и это изучение дает не только формальные знания о структуре пакета,

но и объясняет все основные режимы работы протокола по обработке и передаче IP-дейтаграмм.

Формат заголовка IP-дейтаграммы. IP-дейтаграмма состоит из заголовка и данных. Заголовок дейтаграммы состоит из 32-разрядных слов и имеет переменную длину, зависящую от размера поля “Options”, но всегда кратную 32 битам /19/. За заголовком непосредственно следуют данные, передаваемые в дейтаграмме (рисунок 58).

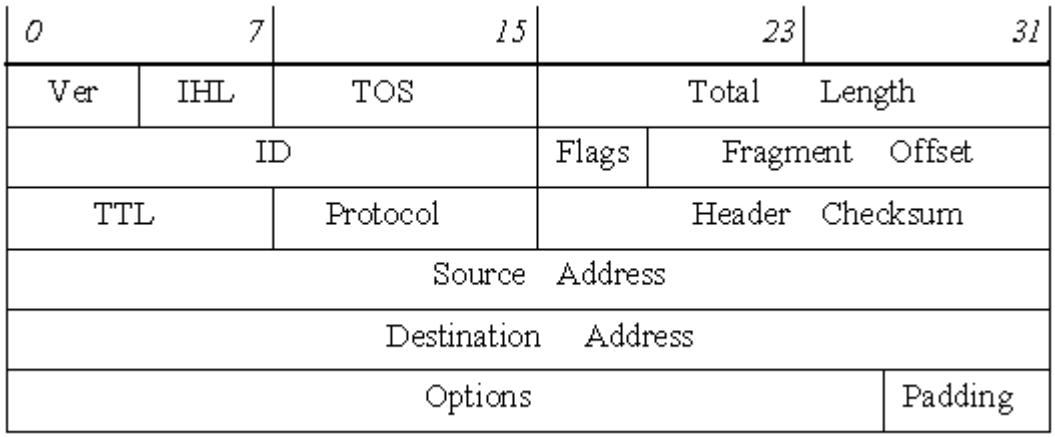


Рисунок 58 - Формат заголовка IP-дейтаграммы

Значения полей заголовка следующие.

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

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

TOS (Type Of Service) (8 бит) - значение поля определяет приоритет дейтаграммы и желаемый тип маршрутизации. Структура байта TOS представлена на рисунке 59.

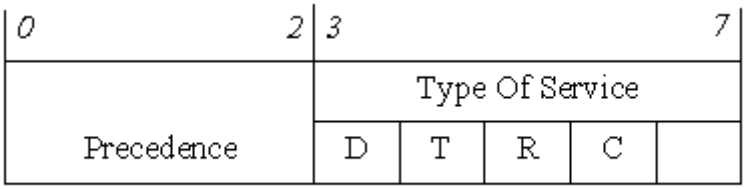


Рисунок 59 - Структура байта TOS

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

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

– D (Delay) - выбор маршрута с минимальной задержкой,

– T (Throughput) - выбор маршрута с максимальной пропускной способностью,

– R (Reliability) - выбор маршрута с максимальной надежностью,

– C (Cost) - выбор маршрута с минимальной стоимостью.

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

Реальный учет приоритетов и выбора маршрута в соответствии со значением байта TOS зависит от маршрутизатора, его программного обеспечения и настроек. Маршрутизатор может поддерживать расчет маршрутов для всех типов TOS, для части или игнорировать TOS вообще. Маршрутизатор может учитывать значение приоритета при обработке всех дейтаграмм или при обработке дейтаграмм, исходящих только из некоторого ограниченного множества узлов сети, или вовсе игнорировать приоритет.

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

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

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

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

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

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

Destination Address (32 бита) - IP-адрес получателя.

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

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

5.2.2 IP-адресация

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

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

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

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

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

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

Все компьютеры объединены в локальную сеть, и имеют локальную IP-адресацию. Пакеты с такой адресацией "путешествовать" в глобальной сети не смогут, т.к. маршрутизаторы их не пропустят.

Поэтому существует шлюз, который преобразовывает пакеты с локальными IP-адресами, давая им свой внешний адрес. И дальше ваши пакеты путешествуют с адресом шлюза (рисунок 60). Таким образом, очень важно

остановится на изучении IP-адреса. IP-адрес является уникальным 32-битным идентификатором IP-интерфейса в Интернет.

IP-адреса принято записывать разбивкой всего адреса по октетам, каждый октет записывается в виде десятичного числа, числа разделяются точками. Например, адрес **10100000010100010000011000011** записывается как **10100000.01010001.00000101.10000011=160.81.5.131.**

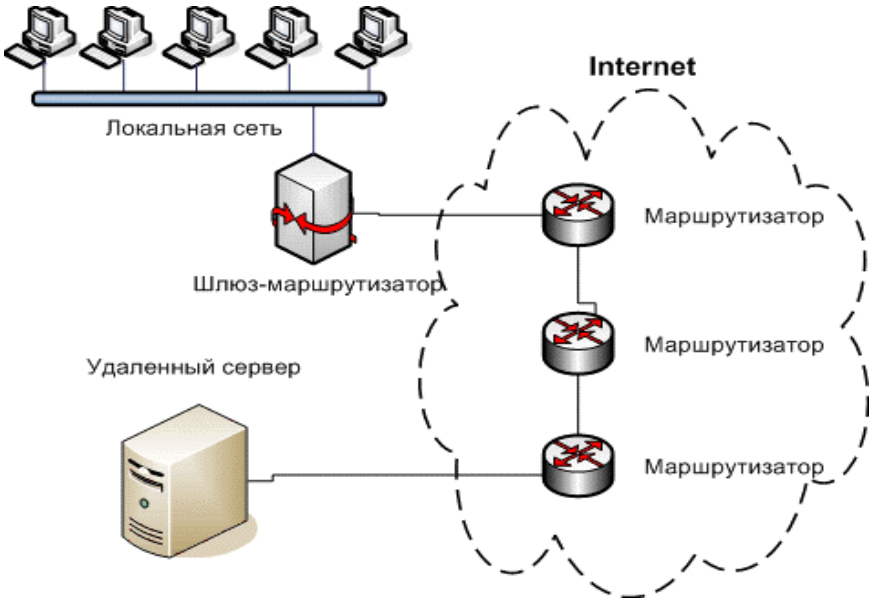


Рисунок 60 – Схема прохождения пакетов из локальной сети к серверу

Положение границы сетевой и хостовой частей (обычно оно характеризуется количеством бит, отведенных на номер сети) может быть различным, определяя различные типы IP-адресов, которые рассматриваются ниже. Старшие биты 4-байтного IP-адреса определяют номер IP-сети, а оставшиеся биты - номер узла.

Классовая модель. IP-адреса разделяются на 5 классов, отличающихся количеством бит в цифровом адресе сети и цифровом адресе узла (таблица 9).

Таблица 9 - Структура IP-адреса

Класс A	0	номер сети	номер узла
Класс B	10	номер сети	номер узла
Класс C	110	номер сети	номер узла
Класс D	1110	групповой адрес	
Класс E	11110	групповой адрес	

Адреса класса A предназначены для использования в больших сетях общего пользования. Адреса класса B предназначены для использования в сетях среднего размера (сети больших компаний, научно-исследовательских институтов, университетов). Адреса класса C предназначены для использования в сетях с небольшим числом компьютеров (сети небольших компаний и фирм). Адреса класса D используют для обращения к группам компьютеров, а адреса класса E - зарезервированы.

– Если адрес начинается с 0, то сеть относят к классу A, и номер сети занимает один байт, остальные 3 байта интерпретируются как номер узла в сети. Сети класса A имеют номера в диапазоне от 1 до 126. (Номер 0 не используется, а номер 127 зарезервирован для специальных целей, о чем будет сказано ниже.) В сетях класса A количество узлов должно быть больше 216 , но не превышать 224.

– Если первые два бита адреса равны 10, то сеть относится к классу B и является сетью средних размеров с числом узлов 28 - 216. В сетях класса B под адрес сети и под адрес узла отводится по 16 битов, то есть по 2 байта.

– Если адрес начинается с последовательности 110, то это сеть класса C с числом узлов не больше 28. Под адрес сети отводится 24 бита, а под адрес узла - 8 битов.

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

– Если адрес начинается с последовательности 11110, то это адрес класса E.

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

Таблица 10 - Характеристики классов адресов

Класс	Наименьший адрес	Наибольший адрес
A	0.1.0.0	126.0.0.0
B	128.0.0.0	191.255.0.0
C	192.0.1.0	223.255.255.0
D	224.0.0.0	239.255.255.255
E	240.0.0.0	247.255.255.255

Среди всех IP-адресов имеется несколько зарезервированных под специальные нужды. Ниже приведена таблица зарезервированных адресов. Особое внимание в таблице уделяется последней строке. Адрес 127.0.0.1 предназначен для тестирования программ и взаимодействия процессов в рамках одного компьютера. В большинстве случаев в файлах настройки этот адрес обязательно должен быть указан, иначе система при запуске может зависнуть.

Наличие "петли" чрезвычайно удобно с точки зрения использования сетевых приложений в локальном режиме для их тестирования и при разработке интегрированных систем. Вообще, зарезервирована вся сеть 127.0.0.0. Эта сеть класса А реально не описывает ни одной настоящей сети.

Некоторые зарезервированные адреса используются для широковещательных сообщений. Например, *номер сети* (строка 2) используется для посылки сообщений этой сети (т.е. сообщений всем компьютерам этой сети). Адреса, содержащие все единицы, используются для широковещательных посылок (для запроса адресов, например).

Таблица 11 - Выделенные IP-адреса

IP-адрес	Значение
все нули	данный узел сети
номер сети | все нули	данная IP-сеть
все нули | номер узла	узел в данной (локальной) сети
все единицы	все узлы в данной локальной IP-сети
номер сети | все единицы	все узлы указанной IP-сети
127.0.0.1	"петля"

Использование масок и подсетей. Часто администраторы сетей испытывают неудобства, из-за того, что количество централизовано выделенных им номеров сетей недостаточно для того, чтобы структурировать сеть надлежащим образом, например, разместить все слабо взаимодействующие компьютеры по разным сетям.

В такой ситуации возможны два пути. Первый из них связан с получением от NIC дополнительных номеров сетей. Второй способ, употребляющийся более часто, связан с использованием так называемых *масок*, которые позволяют разделять одну сеть на несколько сетей.

Маска - это число, двоичная запись которого содержит единицы в тех разрядах, которые должны интерпретироваться как номер сети.

Например, для стандартных классов сетей маски имеют следующие значения:

255.0.0.0 - маска для сети класса А,

255.255.0.0 - маска для сети класса В,

255.255.255.0 - маска для сети класса С.

В масках, которые использует администратор для увеличения числа сетей, количество единиц в последовательности, определяющей границу номера сети, не обязательно должно быть кратным 8, чтобы повторять деление адреса на байты.

Пусть, например, маска имеет значение 255.255.192.0 (11111111 11111111 11000000 00000000). И пусть сеть имеет номер 129.44.0.0 (10000001 00101100 00000000 00000000), из которого видно, что она относится к классу В. После наложения маски на этот адрес число разрядов, интерпретируемых как номер

сети, увеличилось с 16 до 18, то есть администратор получил возможность использовать вместо одного, централизованно заданного ему номера сети, четыре:

129.44.0.0 (10000001 00101100 00000000 00000000)
129.44.64.0 (10000001 00101100 01000000 00000000)
129.44.128.0 (10000001 00101100 10000000 00000000)
129.44.192.0 (10000001 00101100 11000000 00000000)

Например, IP-адрес 129.44.141.15 (10000001 00101100 10001101 00001111), который по стандартам IP задает номер сети 129.44.0.0 и номер узла 0.0.141.15, теперь, при использовании маски, будет интерпретироваться как пара:

129.44.128.0 - номер сети, 0.0. 13.15 - номер узла.

Таким образом, установив новое значение маски, можно заставить маршрутизатор по-другому интерпретировать IP-адрес. При этом два дополнительных последних бита номера сети часто интерпретируются как номера подсетей.

Пример. Пусть некоторая сеть относится к классу B и имеет адрес 128.10.0.0. Этот адрес используется маршрутизатором, соединяющим сеть с остальной частью интерсети. И пусть среди всех станций сети есть станции, слабо взаимодействующие между собой. Их желательно было бы изолировать в разных сетях. Для этого сеть можно разделить на две сети, подключив их к соответствующим портам маршрутизатора, и задать для этих портов в качестве маски, например, число 255.255.255.0, то есть организовать внутри исходной сети с централизовано заданным номером две подсети класса C (можно было бы выбрать и другой размер для поля адреса подсети). Извне сеть по-прежнему будет выглядеть, как единая сеть класса B, а на местном уровне это будут две отдельные сети класса C. Приходящий общий трафик будет разделяться местным маршрутизатором между подсетями.

Бесклассовая модель (CIDR). Предположим, в локальной сети, подключаемой к Интернет, находится 2000 компьютеров. Каждому из них требуется выдать IP-адрес. Для получения необходимого адресного пространства нужны либо 8 сетей класса C, либо одна сеть класса B. Сеть класса B вмещает 65534 адреса, что много больше требуемого количества. При общем дефиците IP-адресов такое использование сетей класса B расточительно. Однако если мы будем использовать 8 сетей класса C, возникнет следующая проблема: каждая такая IP-сеть должна быть представлена отдельной строкой в таблицах маршрутов на маршрутизаторах, потому что с точки зрения маршрутизаторов — это 8 абсолютно никак не связанных между собой сетей, маршрутизация дейтаграмм в которые осуществляется независимо, хотя фактически эти IP-сети и расположены в одной физической локальной сети и маршруты к ним идентичны. Таким образом, экономя адресное пространство, мы многократно увеличиваем служебный трафик в сети и затраты по поддержанию и обработке маршрутных таблиц /20/.

С другой стороны, нет никаких формальных причин проводить границу сеть-хост в IP-адресе именно по границе октета. Это было сделано

исключительно для удобства представления IP-адресов и разбиения их на классы. Если выбрать длину сетевой части в 21 бит, а на номер хоста отвести, соответственно, 11 бит, мы получим сеть, адресное пространство которой содержит 2046 IP-адресов, что максимально точно соответствует поставленному требованию. Это будет *одна* сеть, определяемая своим уникальным 21-битным номером, следовательно, для ее обслуживания потребуется только *одна* запись в таблице маршрутов.

Единственная проблема, которую осталось решить: как определить, что на сетевую часть отведен 21 бит? В случае классовой модели старшие биты IP-адреса определяли принадлежность этого адреса к тому или иному классу и, следовательно, количество бит, отведенных на номер сети.

В случае адресации вне классов, с произвольным положением границы сеть-хост внутри IP-адреса, к IP-адресу прилагается 32-битовая маска, которую называют маской сети (netmask) или маской подсети (subnet mask). Сетевая маска конструируется по следующему правилу:

- на позициях, соответствующих номеру сети, биты установлены;
- на позициях, соответствующих номеру хоста, биты сброшены.

Описанная выше модель адресации называется бесклассовой (CIDR - Classless Internet Direct Routing, прямая бесклассовая маршрутизация в Интернет). В настоящее время классовая модель считается устаревшей и маршрутизация и (большей частью) выдача блоков IP-адресов осуществляются по модели CIDR, хотя классы сетей еще прочно удерживаются в терминологии.

Запись адресов в бесклассовой модели. Для удобства записи IP-адрес в модели CIDR часто представляется в виде a.b.c.d / n, где a.b.c.d — IP адрес, n — количество бит в сетевой части.

Пример: **137.158.128.0/17**.

Маска сети для этого адреса: 17 единиц (сетевая часть), за ними 15 нулей (хостовая часть), что в октетном представлении равно

11111111.11111111.10000000.00000000 = 255.255.128.0.

Представив IP-адрес в двоичном виде и побитно умножив его на маску сети, мы получим номер сети (все нули в хостовой части). Номер хоста в этой сети мы можем получить, побитно умножив IP-адрес на инвертированную маску сети.

Пример: IP = **205.37.193.134/26** или, что то же,

IP = **205.37.193.134** netmask = **255.255.255.192**.

Распишем в двоичном виде:

IP = **11001101 00100101 11000111 10000110**

маска = **11111111 11111111 11111111 11000000**

Умножив побитно, получаем номер сети (в хостовой части - нули):

network = 11001101 00100101 11000111 10000000

или, в октетном представлении, 205.37.193.128/26,

или, что тоже, 205.37.193.128 netmask 255.255.255.192.

Хостовая часть рассматриваемого IP адреса равна 000110, или 6. Таким образом, 205.37.193.134/26 адресует хост номер 6 в сети 205.37.193.128/26. В классовой модели адрес 205.37.193.134 определял бы хост 134 в сети класса C

205.37.193.0, однако указание маски сети (или количества бит в сетевой части) однозначно определяет принадлежность адреса к бесклассовой модели.

Очевидно, что сети классов A, B, C в бесклассовой модели представляются при помощи масок, соответственно, 255.0.0.0 (или /8), 255.255.0.0 (или /16) и 255.255.255.0 (или /24).

5.2.3 Протокол ICMP

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

- сообщает узлу-источнику об отказах маршрутизации;
- проверяет способности узлов образовывать повторное эхо в объединенной сети (сообщения Echo и Reply ICMP);
- стимулирует более эффективную маршрутизацию (с помощью сообщений Redirect ICMP - переадресации ICMP);
- информирует узел-источник о том, что некоторая дейтаграмма превысила назначенное ей время существования в пределах данной сети (сообщение Time Exceeded ICMP - "время превышено");
- обеспечивает для новых узлов возможность нахождения маски подсети, используемой в объединенной сети в данный момент.

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

ICMP-сообщения не порождаются при невозможности доставки:

– дейтаграмм, содержащих ICMP-сообщения;

– не первых фрагментов дейтаграмм;

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

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

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

После IP-заголовка следует 32-битное слово с полями “Тип”, “Код” и “Контрольная сумма”. Поля типа и кода определяют содержание ICMP-сообщения. Формат остальной части дейтаграммы зависит от вида сообщения. Контрольная сумма считается так же, как и в IP-заголовке, но в этом случае суммируется содержимое ICMP-сообщения, включая поля “Тип” и “Код”.

5.3 Транспортные протоколы

5.3.1 Протокол управления передачей TCP

Транспортный уровень Internet реализуется TCP (Transmission Control Protocol, Протокол контроля передачи) и протоколом дейтаграмм пользователя

UDP (User Datagram Protocol). TCP обеспечивает транспортировку данных с установлением соединения, в то время как UDP работает без установления соединения.

Протокол TCP предоставляет транспортные услуги, отличающиеся от услуг UDP. Вместо ненадежной доставки дейтаграмм без установления соединений, он обеспечивает гарантированную доставку с установлением соединений в виде байтовых потоков.

Протокол TCP используется в тех случаях, когда требуется надежная доставка сообщений. Он освобождает прикладные процессы от необходимости использовать таймауты и повторные передачи для обеспечения надежности. Внутренняя структура модуля TCP гораздо сложнее структуры модуля UDP.

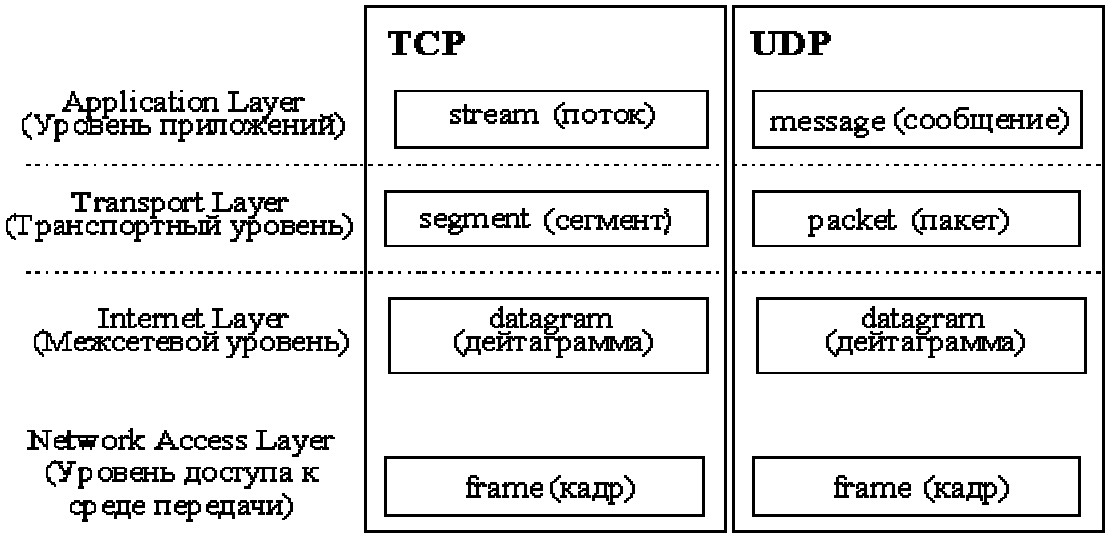


Рисунок 61 – Протоколы транспортного уровня TCP и UDP

TCP - надежный байт-ориентированный (byte-stream) протокол с *установлением соединения*. TCP находится на транспортном уровне стека TCP/IP, между протоколом IP и собственно приложением. Протокол IP занимается пересылкой дейтаграмм по сети, никак не гарантируя доставку, целостность, порядок прибытия информации и готовность получателя к приему данных; все эти задачи возложены на протокол TCP.

При получении дейтаграммы, в поле Protocol которой указан код протокола TCP (6), модуль IP передает данные этой дейтаграммы модулю TCP. Эти данные представляют собой TCP-сегмент, содержащий TCP-заголовок и данные пользователя (прикладного процесса). Модуль TCP анализирует служебную информацию заголовка, определяет, какому именно процессу предназначены данные пользователя, проверяет целостность и порядок прихода данных и подтверждает их прием другой стороне. По мере получения правильной последовательности неискаженных данных пользователя они передаются прикладному процессу /21/.

Ниже основные функции протокола TCP и их реализация рассмотрены более подробно.

Базовая передача данных. Модуль TCP выполняет передачу непрерывных потоков данных между своими клиентами в обоих направлениях. Клиентами TCP являются прикладные процессы, вызывающие модуль TCP при необходимости получить или отправить данные процессу-клиенту на другом узле.

Протокол TCP рассматривает данные клиента как непрерывный не интерпретируемый поток октетов. TCP разделяет этот поток на части для пересылки на другой узел в TCP-сегментах некоторого размера. Для отправки или получения сегмента модуль TCP вызывает модуль IP.

Немедленное отправление данных может быть затребовано процессом-клиентом от TCP-модуля с помощью специальной функции PUSH, иначе TCP сам будет решать, как накапливать и когда отправлять данные клиента или когда передавать клиенту полученные данные.

Обеспечение достоверности. Модуль TCP обеспечивает защиту от повреждения, потери, дублирования и нарушения очередности получения данных.

Для выполнения этих задач все октеты в потоке данных сквозным образом пронумерованы в возрастающем порядке. Заголовок каждого сегмента содержит число октетов данных в сегменте и порядковый номер первого октета той части потока данных, которая пересылается в данном сегменте. Например, если в сегменте пересылаются октеты с номерами от 2001 до 3000, то номер первого октета в данном сегменте равен 2001, а число октетов равно 1000.

Номер первого байта в потоке определяется на этапе установления соединения и обозначается ISN+1. Например, ISN+1=1.

Также для каждого сегмента вычисляется контрольная сумма, позволяющая обнаружить повреждение данных.

При удачном приеме октета данных принимающий модуль посылает отправителю подтверждение о приеме - номер удачно принятого октета. Если в течение некоторого времени отправитель не получит подтверждения, считается, что октет не дошел или был поврежден, и он посылается снова. Этот механизм контроля надежности называется PAR (Positive Acknowledgment with Retransmission). В действительности подтверждение посылается не для одного октета, а для некоторого числа последовательных октетов.

Нумерация октетов используется также для упорядочения данных в порядке очередности и обнаружения дубликатов (которые могут быть посланы из-за большой задержки при передаче подтверждения или потери подтверждения).

Разделение каналов. Протокол TCP обеспечивает работу одновременно нескольких соединений. Каждый прикладной процесс идентифицируется номером порта. Заголовок TCP-сегмента содержит номера портов процесса-отправителя и процесса-получателя. При получении сегмента модуль TCP анализирует номер порта получателя и отправляет данные соответствующему прикладному процессу.

Все распространенные сервисы Интернет имеют стандартизованные номера портов. Например, номер порта сервера электронной почты - 25, сервера FTP - 21.

Совокупность IP-адреса и номера порта называется *сокетом*. Сокет уникально идентифицирует прикладной процесс в Интернет. Например, сокет сервера электронной почты на хосте 194.84.124.4 обозначается как 194.84.124.4.25; часто номер порта отделяется двоеточием.

Управление соединениями. Соединение - это совокупность информации о состоянии потока данных, включающая сокеты, номера посланных, принятых и подтвержденных октетов, размеры окон.

Каждое соединение уникально идентифицируется в Интернет парой сокетов. Соединение характеризуется для клиента именем, которое является указателем на структуру TCB (Transmission Control Block), содержащую информацию о соединении.

Открытие соединения клиентом осуществляется вызовом функции OPEN, которой передается сокет, с которым требуется установить соединение. Функция возвращает имя соединения. Различают два типа открытия соединения: активное и пассивное.

При активном открытии TCP-модуль начинает процедуру установления соединения с указанным сокетом, при пассивном - ожидает, что удаленный TCP-модуль начнет процедуру установления соединения с указанного сокета.

Указание 0.0.0.0:0 в качестве сокета при пассивном открытии означает, что ожидается соединение с любого сокета. Такой способ применяется в демонах - серверах Интернет, которые ждут установления соединения от клиента. Клиент же применяет процедуру активного открытия; сокет при этом формируется из IP-адреса сервера и стандартного номера порта для данного сервиса. Закрытие соединения клиентом производится с помощью функции CLOSE, которой передается имя соединения. Процедура установления соединения происходит следующим образом (рисунок 62).

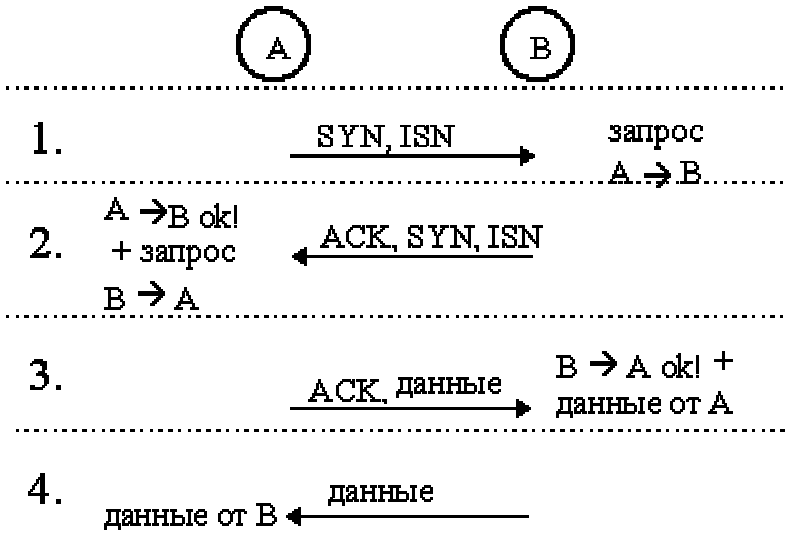


Рисунок 62 – Установка TCP-соединения

Предположим, узел А желает установить соединение с узлом В. Первый отправляемый из А в В TCP-сегмент не содержит полезных данных, а служит для установления соединения. В его заголовке установлен бит SYN, означающий запрос связи, и содержится ISN (Initial Sequence Number - начальный номер последовательности) - число, начиная с которого узел А будет нумеровать отправляемые октеты (например, 0). В ответ на получение такого сегмента узел В откликается посылкой TCP-сегмента, в заголовке которого установлен бит ACK, подтверждающий установление соединения для получения данных от узла А. Так как протокол TCP обеспечивает полнодуплексную передачу данных, то узел В в этом же сегменте устанавливает бит SYN, означающий запрос связи для передачи данных от В к А, и передает свой ISN (например, 0). Полезных данных этот сегмент также не содержит. Третий TCP-сегмент в сеансе посылается из А в В в ответ на сегмент, полученный из В. Так как соединение А -> В можно считать установленным (получено подтверждение от В), то узел А включает в свой сегмент полезные данные, нумерация которых начинается с номера ISN(A)+1. Данные нумеруются по количеству отправленных октетов. В заголовке этого же сегмента узел А устанавливает бит ACK, подтверждающий установление связи В -> А, что позволяет хосту В включить в свой следующий сегмент полезные данные для А /22/.

Сеанс обмена данными заканчивается процедурой разрыва соединения, которая аналогична процедуре установки, с той разницей, что вместо SYN для разрыва используется служебный бит FIN (“данных для отправки больше не имею”), который устанавливается в заголовке последнего сегмента с данными, отправляемого узлом.

Формат заголовока TCP-сегмента. TCP-сегмент состоит из заголовка и данных. Заголовок сегмента состоит из 32-разрядных слов и имеет переменную длину, зависящую от размера поля Options, но всегда кратную 32 битам. За заголовком непосредственно следуют данные - часть потока данных пользователя, передаваемая в данном сегменте. Формат заголовка представлен на рисунке 63.

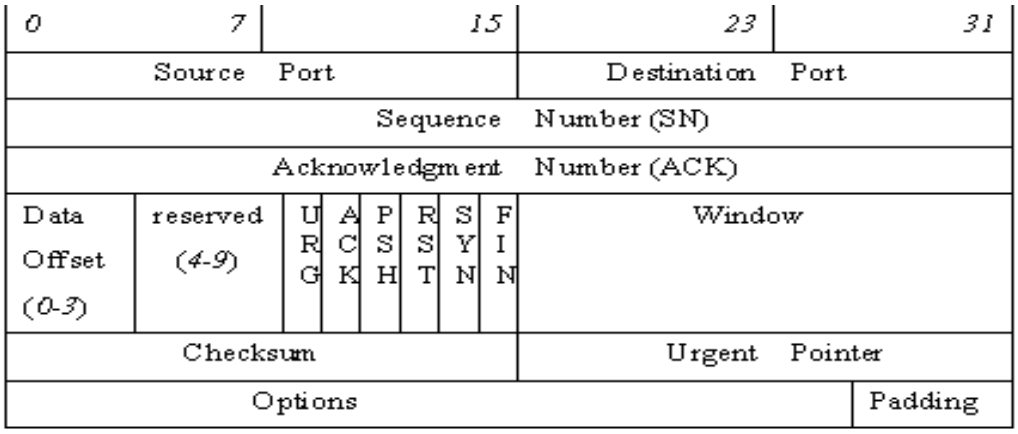


Рисунок 63 – Формат заголовка TCP-сегмента

Значения полей заголовка следующие.

Source Port (16 бит), Destination Port (16 бит) - номера портов процесса-отправителя и процесса-получателя соответственно.

Sequence Number (SN) (32 бита) - порядковый номер первого октета в поле данных сегмента среди всех октетов потока данных для текущего соединения, то есть если в сегменте пересылаются октеты с 2001-го по 3000-й, то SN=2001. Если в заголовке сегмента установлен бит SYN (фаза установления соединения), то в поле SN записывается начальный номер (ISN), например, 0. Номер первого октета данных, посылаемых после завершения фазы установления соединения, равен ISN+1. Acknowledgment Number (ACK) (32 бита) - если установлен бит ACK, то это поле содержит порядковый номер октета, который отправитель данного сегмента желает получить. Это означает, что все предыдущие октеты (с номерами от ISN+1 до ACK-1 включительно) были успешно получены.

Data Offset (4 бита) - длина TCP-заголовка в 32-битных словах.

Reserved (6 бит) - зарезервировано; заполняется нулями.

Control Bits (6 бит) - управляющие биты; активным является положение “бит установлен”.

URG - поле срочного указателя (Urgent Pointer) задействовано;

ACK - поле номера подтверждения (Acknowledgment Number) задействовано;

PSH - осуществить “проталкивание” - если модуль TCP получает сегмент с установленным флагом PSH, то он немедленно передает все данные из буфера приема процессу-получателю для обработки, даже если буфер не был заполнен;

RST - перезагрузка текущего соединения;

SYN - запрос на установление соединения;

FIN - нет больше данных для передачи.

Window (16 бит) - размер окна в октетах.

Checksum (16 бит) - контрольная сумма, представляет собой 16 бит, дополняющие биты в сумме всех 16-битовых слов сегмента (само поле контрольной суммы перед вычислением обнуляется). Контрольная сумма, кроме заголовка сегмента и поля данных, учитывает 96 бит псевдозаголовка, который для внутреннего употребления ставится перед TCP-заголовком. Этот псевдозаголовок содержит IP-адрес отправителя (4 октета), IP-адрес получателя (4 октета), нулевой октет, 8-битное поле "Протокол", аналогичное полю в IP-заголовке, и 16 бит длины TCP сегмента, измеренной в октетах. Такой подход обеспечивает защиту протокола TCP от ошибшихся в маршруте сегментов. Информация для псевдозаголовка передается через интерфейс "Протокол TCP/межсетевой уровень" в качестве аргументов или результатов запросов от протокола TCP к протоколу IP.

Urgent Pointer (16 бит) - используется для указания длины срочных данных, которые размещаются в начале поля данных сегмента. Указывает смещение октета, следующего за срочными данными, относительно первого октета в сегменте. Например, в сегменте передаются октеты с 2001-го по 3000-й, при этом первые 100 октетов являются срочными данными, тогда Urgent

Pointer = 100. Протокол TCP не определяет, как именно должны обрабатываться срочные денные, но предполагает, что прикладной процесс будет предпринимать усилия для их быстрой обработки. Поле Urgent Pointer задействовано, если установлен флаг URG.

Options - поле переменной длины; может отсутствовать или содержать одну опцию или список опций, реализующих дополнительные услуги протокола TCP. Опция состоит из октета "Тип опции", за которым могут следовать октет "Длина опции в октетах" и октеты с данными для опции.

Стандарт протокола TCP определяет три опции (типы 0,1,2).

Опции типов 0 и 1 ("Конец списка опций" и "Нет операции" соответственно) состоят из одного октета, содержащего значение типа опции. При обнаружении в списке опции "Конец списка опций" разбор опций прекращается, даже если длина заголовка сегмента (Data Offset) еще не исчерпана. Опция "Нет операции" может использоваться для выравнивания между опциями по границе 32 бит.

Опция типа 2 "Максимальный размер сегмента" состоит из 4 октетов: одного октета типа опции (значение равно 2), одного октета длины (значение равно 4) и двух октетов, содержащих максимальный размер сегмента, который способен получать TCP-модуль, отправивший сегмент с данной опцией. Опцию следует использовать только в SYN-сегментах на этапе установки соединения.

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

5.3.2 Протокол дейтаграмм пользователя UDP

Протокол UDP (User Datagram Protocol, протокол пользовательских дейтаграмм) используется в тех случаях, когда мощные средства обеспечения надежности протокола TCP не требуются. Реализация UDP намного проще, чем TCP.

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

Пользовательские данные, поступившие от прикладного уровня, предваряются UDP-заголовком, и сформированный таким образом UDP-пакет отправляется на межсетевой уровень. UDP-заголовок состоит из двух 32-битных слов (рисунок 64).

Заголовок UDP имеет четыре поля:

– порт источника (source port) - те же функции, что и в заголовке TCP;

– порт пункта назначения (destination port) - те же функции, что и в заголовке TCP;

– длина (length) - длина заголовка UDP и данных;

– контрольная сумма UDP (checksum UDP) - обеспечивает проверку целостности пакета (факультативная возможность).

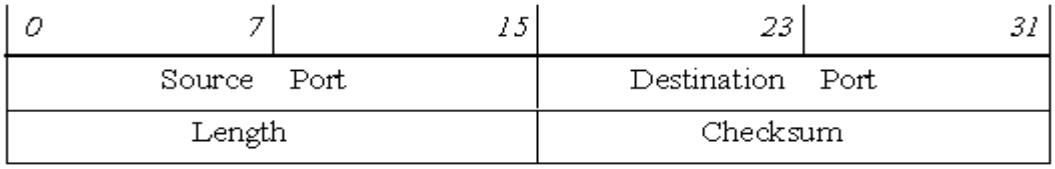


Рисунок 64 - UDP-заголовок

Контрольная сумма вычисляется таким же образом, как и в TCP-заголовке; если UDP-пакет имеет нечетную длину, то при вычислении контрольной суммы к нему добавляется нулевой октет.

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

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

Если проверка контрольной суммы выявила ошибку или если процесса, подключенного к требуемому порту, не существует, пакет игнорируется. Если пакеты поступают быстрее, чем модуль UDP успевает их обрабатывать, то поступающие пакеты также игнорируются. Протокол UDP не имеет никаких средств подтверждения безошибочного приема данных или сообщения об ошибке, не обеспечивает приход сообщений в порядке отправки, не производит предварительного установления сеанса связи между прикладными процессами, поэтому он является ненадежным протоколом без установления соединения. Если приложение нуждается в подобного рода услугах, оно должно использовать на транспортном уровне протокол TCP.

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

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

Порты. Взаимодействие между прикладными процессами и модулем UDP осуществляется через UDP-порты. Порты нумеруются начиная с нуля.

Прикладной процесс, предоставляющий некоторые услуги другим прикладным процессам (сервер), ожидает поступления сообщений в порт, специально выделенный для этих услуг. Сообщения должны содержать запросы на предоставление услуг. Они отправляются процессами-клиентами.

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

Данные, отправляемые прикладным процессом через модуль UDP, достигают места назначения как единое целое. Например, если процессотправитель производит 5 записей в UDP-порт, то процесс-получатель должен будет сделать 5 чтений. Размер каждого записанного сообщения будет совпадать с размером каждого прочитанного. Протокол UDP сохраняет границы сообщений, определяемые прикладным процессом. Он никогда не объединяет несколько сообщений в одно и не делит одно сообщение на части.

По номеру порта транспортные протоколы определяют, какому приложению передать содержимое пакетов /23/.

5.4 Связь протоколов сетевого и транспортного уровней

TCP/IP - собирательное название для стека сетевых протоколов разных уровней, используемых в Интернет. Термин "TCP/IP" обозначает технологию межсетевого взаимодействия на основе семейства протоколов TCP и IP (рисунок 65).

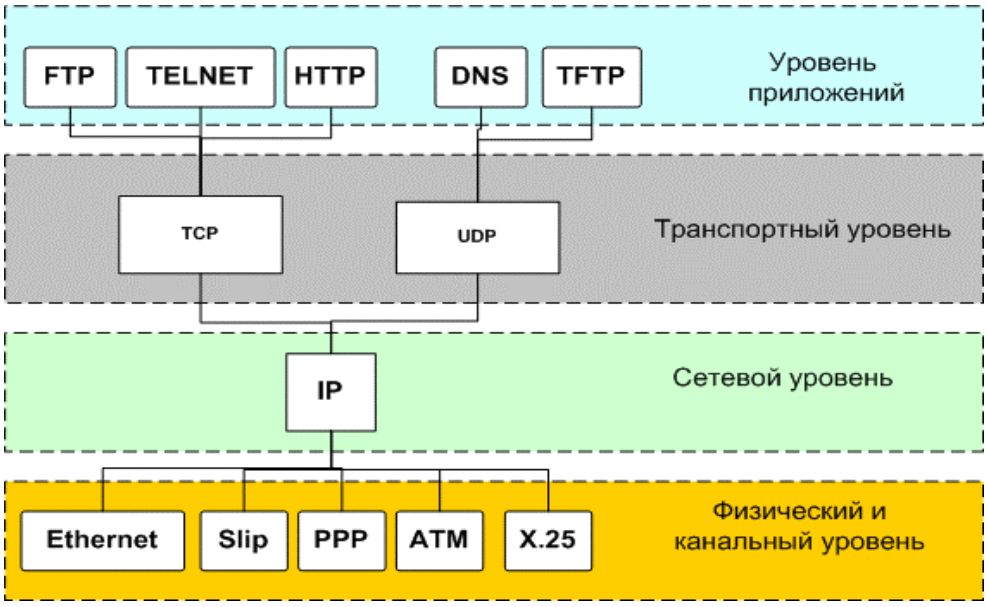


Рисунок 65 - Связь протоков в узле TCP/IP

В это семейство входят протоколы UDP, ARP, ICMP, TELNET, FTP и многие другие. Работа TCP/IP заключается в следующем. Протокол TCP разбивает информацию на порции и нумерует все порции, чтобы при получении можно было правильно собрать информацию. Далее с помощью протокола IP все части передаются получателю, где с помощью протокола TCP проверяется, все ли части получены. Так как отдельные части могут путешествовать по Интернет самыми разными путями, то порядок прихода частей может быть нарушен. После получения частей TCP располагает их в нужном порядке и собирает в единое целое.

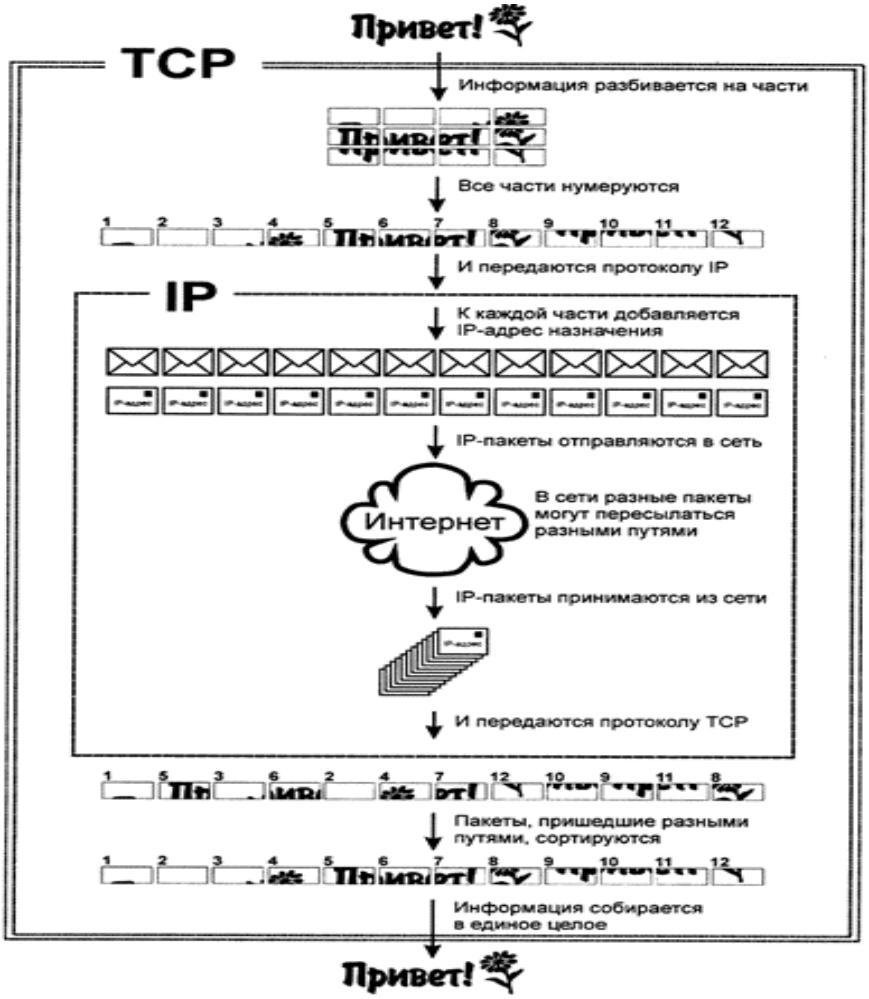


Рисунок 66 - Схема функционирования протокола TCP/IP

Для протокола TCP не имеет значения, какими путями информация путешествует по Интернет. Этим занимается протокол IP**.** К каждой полученной порции информации протокол IP добавляет служебную информацию, из которой можно узнать адреса отправителя и получателя информации. Если следовать аналогии с почтой, то данные помещаются в конверт, на котором пишется адрес получателя. Далее протокол IP, так же как и обычная почта, обеспечивает доставку всех конвертов получателю. При этом скорость и пути прохождения разных конвертов могут быть различными. Если при путешествии отдельного конверта наблюдались помехи, и информация пришла искаженной, следует повторный запрос об отправке искаженной части до тех пор, пока она

не будет принята без искажений (в этом еще один плюс приема–передачи информации порциями).

Архитектура протоколов TCP/IP предназначена для объединенной сети, состоящей из соединенных друг с другом шлюзами отдельных разнородных пакетных подсетей, к которым подключаются разнородные машины. Каждая из подсетей работает в соответствии со своими специфическими требованиями и имеет свою природу средств связи. Однако предполагается, что каждая подсеть может принять пакет информации (данные с соответствующим сетевым заголовком) и доставить его по указанному адресу в этой конкретной подсети. Не требуется, чтобы подсеть гарантировала обязательную доставку пакетов и имела надежный сквозной протокол. Таким образом, две машины, подключенные к одной подсети могут обмениваться пакетами.

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

Использование во всех узлах и маршрутизаторах межсетевого протокола IP решает проблему доставки пакетов. Таким образом, обеспечивается дейтаграммный сервис на межсетевом уровне Internet. Этот уровень обеспечивает возможность стандартизации протоколов верхних уровней и является основой архитектуры TCP/IP.

5.4.1 Структура связей протокольных модулей

Логическая структура сетевого программного обеспечения, реализующего протоколы семейства TCP/IP в каждом узле сети internet, изображена на рисунке 67.

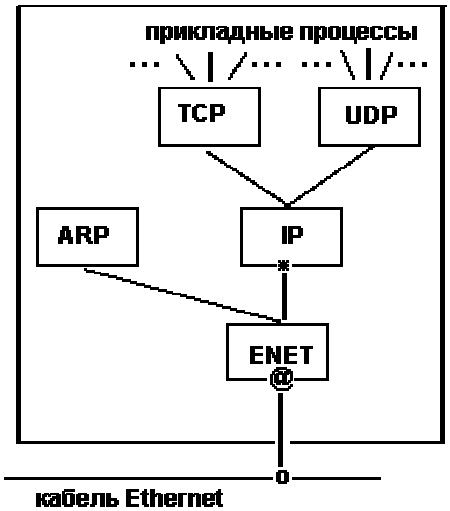


Рисунок 67- Структура протокольных модулей в узле сети TCP/IP

Прямоугольники обозначают обработку данных, а линии, соединяющие прямоугольники, - пути передачи данных. Горизонтальная линия внизу рисунка обозначает кабель сети Ethernet, которая используется в качестве примера физической среды; "o" - это трансивер.

Знак "*" – обозначает IP-адрес, а "@" - адрес узла в сети Ethernet (Ethernet-адрес). Понимание этой логической структуры является основой для понимания всей технологии internet.

Рассмотрим потоки данных, проходящие через стек протоколов, изображенный на рисунке 67. В случае использования протокола TCP, данные передаются между прикладным процессом и модулем TCP. Типичным прикладным процессом, использующим протокол TCP, является модуль FTP. Стек протоколов в этом случае будет FTP/TCP/IP/ENET. При использовании протокола UDP, данные передаются между прикладным процессом и модулем UDP. Например, SNMP пользуется транспортными услугами UDP. Его стек протоколов выглядит так: SNMP/UDP/IP/ENET.

Модули TCP, UDP и драйвер Ethernet являются мультиплексорами n x 1. Действуя как мультиплексоры, они переключают несколько входов на один выход. Они также являются демультиплексорами 1 x n. Как демультиплексоры, они переключают один вход на один из многих выходов в соответствии с полем типа в заголовке протокольного блока данных (рисунок 68).

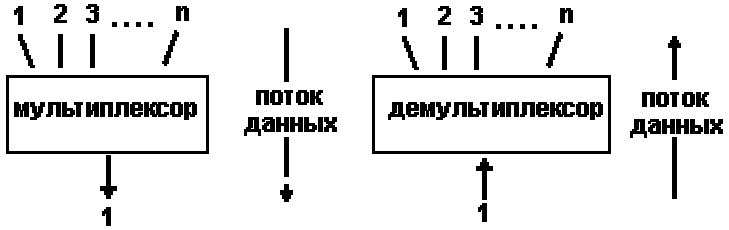


Рисунок 68 - Мультиплексор n x 1 и демультиплексор 1 x n

Когда Ethernet-кадр попадает в драйвер сетевого интерфейса Ethernet, он может быть направлен либо в модуль ARP (Address Resolution Protocol адресный протокол), либо в модуль IP (Internet Protocol - межсетевой протокол). На то, куда должен быть направлен Ethernet-кадр, указывает значение поля типа в заголовке кадра.

Если IP-пакет попадает в модуль IP, то содержащиеся в нем данные могут быть переданы либо модулю TCP, либо UDP, что определяется полем "протокол" в заголовке IP-пакета.

Если UDP-дейтаграмма попадает в модуль UDP, то на основании значения поля "порт" в заголовке дейтаграммы определяется прикладная программа, которой должно быть передано прикладное сообщение. Если TCP-сообщение попадает в модуль TCP, то выбор прикладной программы, которой должно быть передано сообщение, осуществляется на основе значения поля "порт" в заголовке TCP-сообщения.

Мультиплексирование данных в обратную сторону осуществляется довольно просто, так как из каждого модуля существует только один путь вниз.

Каждый протокольный модуль добавляет к пакету свой заголовок, на основании которого машина, принявшая пакет, выполняет демультиплексирование. Данные от прикладного процесса проходят через модули TCP или UDP, после чего попадают в модуль IP и оттуда - на уровень сетевого интерфейса.

Хотя технология internet поддерживает много различных сред передачи данных предполается использование Ethernet, так как именно эта среда чаще всего служит физической основой для IP-сети. Машина на рисунке 67 имеет одну точку соединения с Ethernet. Шестибайтный Ethernet-адрес является уникальным для каждого сетевого адаптера и распознается драйвером.

Машина имеет также четырехбайтный IP-адрес. Этот адрес обозначает точку доступа к сети на интерфейсе модуля IP с драйвером. IP-адрес должен быть уникальным в пределах всей сети Internet. Работающая машина всегда знает свой IP-адрес и Ethernet-адрес.

Работа с несколькими сетевыми интерфейсами. Одна машина может быть подключена одновременно к нескольким сегментам сети (средам передачи данных) /24/. Например, машина на рисунке 69 имеет два сетевых интерфейса Ethernet и, следовательно, 2 Ethernet-адреса. Из рисунка также видно, что эта машина имеет также 2 IP-адреса. Также видно, что в рассматриваемом случае модуль IP выполняет более сложную функцию - сложнее, чем в первом примере, так как осуществляет мультиплексирование входных и выходных данных в обоих направлениях и может передавать (ретранслировать) данные между сетями, модуль IP выполняет функции мультиплексора n x m и демультиплексора m x n (рисунок 69).

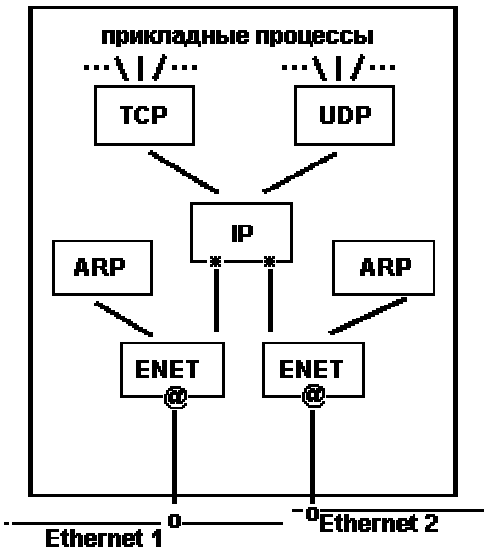


Рисунок 69 - Узел сети TCP/IP с двумя сетевыми интерфейсами

Такая функция, называется маршрутизацией. Данные, поступившие через один сетевой интерфейс, могут быть ретранслированы через другой сетевой интерфейс. Ретранслируемый пакет не поступает в модули TCP или UDP. Если модули TCP и UDP отсутствуют, то мы имеем дело с *машиной-*

маршрутизатором. Если модули TCP и UDP имеются, то *это рабочая станция.*

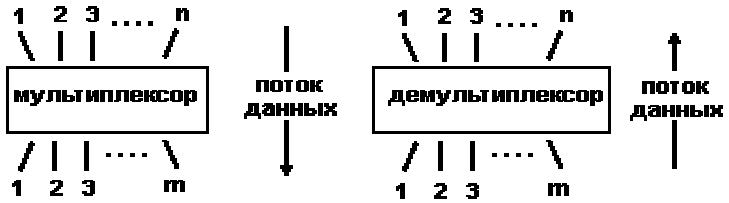


Рисунок 70 - Мультиплексор n x m и демультиплексор m x n

Таким образом, он осуществляет мультиплексирование входных и выходных данных в обоих направлениях. Модуль IP в данном случае сложнее, чем в первом примере, так как может передавать данные между сетями. Данные могут поступать через любой сетевой интерфейс и быть ретранслированы через любой другой сетевой интерфейс. Процесс передачи пакета в другую сеть называется ретрансляцией IP-пакета. Машина, выполняющая ретрансляцию, называется шлюзом.

5.5 Контрольные вопросы

1) Классификация протоколов.
2) Сетевые протоколы
3) Адресация в сети Internet. IP-адреса.
4) Транспортные протоколы.
5) Особенности TCP/IP.

5.6 Тесты

1) На каком уровне стека протокола TCP/IP находится протокол IP?
   а) представительский;
   б) сеансовый;
   в) транспортный;
   г) межсетевой.

2) Основная задача, решаемая протоколом IP:
   а) маршрутизация;
   б) добавление заголовка;
   в) анализ правильности доставки.

3) Из какого количества байт состоит IP-адрес?
   а) 1;
   б) 2;
   в) 3;
   г) 4.

4) MAC-адрес – это:

а) адрес, назначаемый динамически при входе в сеть;

б) адреса, назначаемые производителями оборудования и являющиеся уникальными;

в) адрес, выбираемый пользователем при входе в сеть.

5) Старшие биты 4-байтного IP-адреса определяют:

а) номер сети;

б) номер подсети;

в) номер хоста;

г) MAC-адрес.

6) Сколько классов IP-адресов вы знаете?

а) 2;

б) 3;

в) 5;

г) 9.

7) Выберите IP-адрес, соответствующий интерфейсу обратной связи:

а) 192.168.1.123;

б) 127.0.0.1;

в) 13.45.12.1;

г) 1.1.1.1;

8) Выберите IP-адрес, соответствующий всей сети целиком:

а) 192.168.1.1;

б) 127.0.0.0;

в) 255.255.255.255;

г) 0.0.0.0.

9) Выберите IP-адрес, соответствующий широковещательной передаче:

а) 192.168.1.1;

б) 127.0.0.0;

в) 255.255.255.255;

г) 1.1.1.1.

10) Протокол ICMP:

а) обеспечивает обратную связь в виде диагностических сообщений, посылаемых отправителю при невозможности доставки его дейтаграммы и в других случаях;

б) отвечает непосредственно за передачу данных;

в) отвечает за корректное принятие данных;

г) посылает широковещательное сообщение.

11) К какому из перечисленных протоколов подходит определение

“Дейтаграммный протокол транспортного уровня”?

а) TFTP;
б) SPX;
в) TCP;
г) UDP.

12) Протокол TCP работает:
а) с установлением соединения;
б) без установления соединения.

13) Сокет – это:
а) IP-адрес;
б) номер сети, входящий в IP-адрес;
в) порт ПК, находящегося в сети;
г) IP-адрес и номер порта.

14) Команда PING используется:
а) для просмотра локального MAC-адреса;
б) для просмотра IP-адреса ПК;
в) для проверки соединения с удаленным хостом;
г) для отправки широковещательного сообщения.

15) DNS – это:
а) удаленный файл-сервер;
б) сервер доменных имен;
в) мощный поисковый сервер.

6 Информационные сервисы Internet

6.1 История развития сети Internet

Интернет - это сеть сетей компьютеров. Сети посредством компьютеров соединяют и объединяют людей. Большой успех Интернета обуславливается не физическим соединении компьютеров, а соединением и объединением людей.

Джон Десембер, один из наиболее популярных в мире авторов статей и книг про Internet дает следующее определение: Интернет - это всемирная кооперативно управляемая совокупность компьютерных сетей, обменивающихся информацией с помощью протоколов TCP/IP.

Этап I. Создание ARPANET. В 1958 году, в ответ на запуск советского спутника, США создают организацию ARPA. Усилия организации,

направленные на исследования в области компьютерных технологий и способов передачи информации, возглавил тогда д-р Ликлайдер. Обработка, хранение, передача информации — все эти процессы тогда выполнялись на перфокартах, что существенно усложняло весь процесс исследований и расчетов. Поэтому первоначальная задача перед доктором Ликлайдером стояла в изменении самого технологического процесса способов передачи информации /25/.

В 1963 году Ликлайдер начал тесное сотрудничество с Ларри Робертсом, который был признанным специалистом в области компьютерной графики. В ходе дискуссий было решено организовать сеть передачи данных, основанную на архитектуре с распределенными параметрами. Главное ее преимущество - высокая степень защищенности в случае поражения отдельных частей сети, что хорошо иллюстрирует рисунок 71, позиция С.

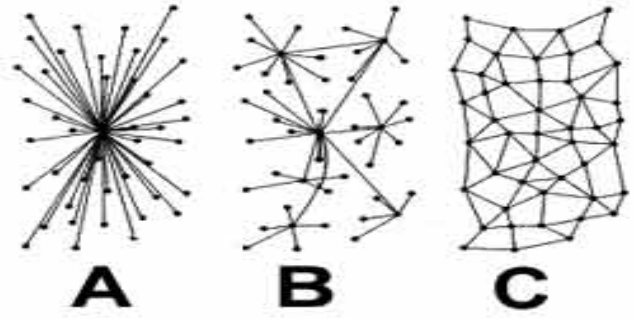


Рисунок 71 - Возможные архитектуры исследуемой сети передачи данных

Организация ARPA перерастает в новую организацию под названием **ARPANET**. Но и ARPANET, поначалу, была "без форм и оболочки". Заслуга Ликлайдера заключалась, прежде всего, в том, что он первоначально сформулировал концепцию сети как средство коммуникации людей посредством передачи информации. Ларри Робертс утверждает, что именно такое видение сети Ликлайдером и его знание "как сделать" помогло созданию ARPANET, а затем и всех других сетей-потомков. Ликлайдер говорил: *"Идет превращение компьютера из арифметического процессора в средство общения"*. Ликлайдер обуславливал свою историческую миссию, понимая, что, изменив представление о компьютере, люди открывают его новые возможности.

Весной 1967 года в Университете города Мичигана состоялась ежегодная встреча "исследователей принципа" обработки и передачи информации. Ее цель была скоординировать дальнейшие шаги в развитии сетей.

На этой встрече ARPA объявило тендер для организаций, способных решить два вопроса:

1. Сконструировать базовую сеть передачи данных, состоящую из телефонных линий и узлов коммутации, чьи надежность, вместимость и стоимость содействовали бы распределению ресурсов в сети.

2. Понимать и выполнять протоколы и процедуры внутри операционных систем каждого соединенного компьютера, для того чтобы сделать возможным включение новой подсети в имеющуюся.

Этап II. Развитие ARPANET. Всю первую половину 1969 года продолжались работы над иерархией протоколов передачи данных. Суть проблемы состояла в разделении на уровни взаимодействия частей компьютеров в сети (аппаратной, программной частей, уровень модема и т.д.). Также система должна была поддерживать протокол удаленного доступа и запуска программ (telnet) и передачи файлов (ftp).

Одновременно при Калифорнийском университете был создан Центр сетевых измерений. Было решено соединить те исследовательские центры, которые были вовлечены в создание ARPANET, этим был положен конец всем попутным и отпочковавшимся исследованиям по сетям. Один из таких узлов должен был быть максимально удален, чтобы проверить систему в максимальных режимах. Что касается аппаратной части - "железа", то остановились на 16-разрядном мини-компьютере Honeywell DDP-316 с 12 Кбайт памяти. Линии связи емкостью 56 Кбайт/с были арендованы у телефонной компании AT&T. Программное обеспечение состояло из соединений IMP — host, IMP — IMP — протокол, протокол IMP-отправитель — IMP-получатель (IMP-s-IMP-r), как показано на рисунке 72.

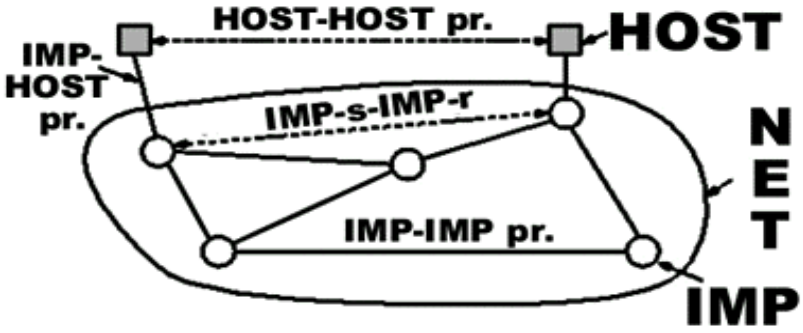


Рисунок 72 - Схема построения сети ARPANET

На сентябре 1969 года под руководством Леонарда Клейнрока был установлен IMP-процессор в Калифорнийском университете. Центр сетевых измерений находился там же. Установленные по всем университетам компьютеры относились к разным производителям и маркам и не были совместимыми. Поэтому в конце 1971 года Ларри Робертс решил всех научить "говорить на одном языке". По принципу Ликлайдера он решил убедить компьютерную общественность в необходимости единых стандартов. В октябре 1972 года было решено провести Международную конференцию по компьютерной связи, на которой Ларри Робертс организовал общественную демонстрацию ARPANET.

Демонстрация имела большой успех, особенно среди представителей AT&T (в те времена монополист в области телекоммуникаций), которые первоначально скептически относились к возможностям ARPANET. Корпорации заинтересовались разработками ARPA, увидев их практическую реализацию и коммерческую выгоду. Началась разработка устраивающего бы всех протокола. Так была предопределена необходимость протокола TCP/IP.

Этап III. Преобразование ARPANET. Весной 1973 года Винт Серф и

Боб Кан задумались о том, как бы им соединить новообразовывающиеся сети с ARPANET, ведь к тому времени таковые уже были (например, SATNET). Эти сети имели другие принципы организации, использовали другие протоколы, были предназначены для решения других задач. Серф и Кан предложили протоколы глобальной связи для сетей с пакетной коммутацией. Фактически, предлагался новый протокол, суть которого была в том, чтоб создать конверт, в который "завернута" часть письма (эту часть письма внутри конверта было предложено назвать "дейтаграммой"). Сетям нужно было только понимать "надпись" на конверте, чтобы передать его в место назначения, а до содержания его им дела не было. Если конверт не доходил до "адресата", то должен быть выслан новый конверт. Этот протокол позволил "разговаривать" совершенно разным сетям и был назван протоколом контроля передачи или TCP.

В июле 1977 Серф и Кан впервые продемонстрировали передачу данных с использованием TCP по трем различным сетям. Пакет прошел по следующему маршруту: Сан-Франциско — Лондон — Университет Южной Калифорнии. В конце своего путешествия пакет проделал 150 тысяч км, не потеряв ни одного бита. В 1978 году Серф, Постел и Дэни Кохэн решили выделить в TCP две отдельные функции: TCP и протокол Интернета (IP). TCP был ответственен за разбивку сообщения на дейтаграммы и соединение их в конечном пункте отправки. IP отвечал за передачу (с контролем получения) отдельных дейтаграмм. 1 января 1983 года ARPANET перешла на новый протокол. Этот день принято считать официальной датой рождения Интернета.

В начале 80-ых сети начали бурно развиваться. Можно отметить самые масштабные из них: CSNET (компьютерная научно-исследовательская сеть), CDNET (канадская сеть), MILNET (сеть МО США) и самая большая NSFNET (национальная научная сеть).

В 1977 году ARPANET состояла из 111 хост-компьютеров, а уже в 1983 году — из 4000, которые располагались по всем США, была налажена спутниковая связь с Гавайями и Европой.

В 1989 Интернет стал набирать обороты, все больше сеть использовалась в коммерческих целях, все менее в научных. К тому же, упомянутая NSFNET была ориентирована именно на научную аудиторию, эта научная сеть была быстрее ARPANET, в ней было больше компьютеров. В конце концов, в ARPA решили умертвить свое детище, успевшее прожить 22 года, а входящие в ARPANET компьютеры передать NSFNET. Данную миссию, отключая один за другим от ARPANET компьютеры, сделал Марк Пуллен.

Уникальным явлением было рождение документации под общим названием Request For Comments (RFC — дословно "просьба комментариев"). Идея была рождена еще первой Сетевой рабочей группой (NWG), а именно Стивом Крокером. Любой из членов группы приносил любые свои теоретические наработки касательно сетей, а другие ручкой на полях вносили комментарии, следующая версия наработок издавалась с комментариями, потом добавлялись все новые и новые комментарии. Так обсуждались теории, результатом было общее сформированное всей группой видение какой-либо

теории. Практика "Просьбы комментариев" прижилась в научной среде, что-то подобное перекочевало в Интернет в виде конференций с обсуждениями.

6.2 Основные инструменты Internet

Так как Internet имеет различные уникальные ресурсы, то для их использования созданы разные инструменты, основными из которых являются:

1) электронная почта (e-mail) – где, в отличие от обычной почты, сообщения передаются практически моментально; при этом не имеет никакого значения, находится ли адресат за соседним столом или же в другом полушарии. По электронной почте можно передавать не только текст, но и компьютерные программы, графическое изображение, компьютерные игры и многое другое. Через электронную почту можно отправить факс - хотя и только текстовый.

2) списки рассылки (mailing lists) - подписавшись на такой список можно получать в свой электронный почтовый ящик информацию по интересующей теме.

3) электронные конференции (newsgroups, electronic conferences) - это своего рода "электронные доски объявлений" или "рабочие группы". От обычных конференций они отличаются тем, что их участники могут одновременно "присутствовать" сразу на нескольких конференциях, а если конференция международная, то нет необходимости приезжать на нее.

4) FTP - протокол передачи файлов, позволяет абонентам получать на свой персональный компьютер файлы с удаленных компьютеров (и отправлять, при необходимости, в ответ собственные файлы).

5) Telnet - протокол, позволяющий работать с ресурсами удаленного компьютера (в т.ч. выполнять различные программы, проверять на нем свою почту, получать доступ к базам данных и т.п.).

6) Gopher - инструмент для поиска и получения информации с помощью перемещения по системам вложенных меню.

7) WWW (World Wide Web) - средство работы с Internet, позволяющее легко добраться до нужного ресурса сети Internet. Это всемирная мультимедийная среда с мощнейшими средствами поиска и представления информации, в ней доступен и звук, и видео, и элементы виртуальной реальности.

8) On-line Databases - поиск данных в режиме диалога реального времени в различных базах данных, поддерживаемых на компьютерах Internet.

и многие другие.

6.3 Система доменных имен

Числовая адресация удобна для машинной обработки таблиц маршрутов, но совершенно не приемлема для использования ее человеком. Запомнить наборы цифр гораздо труднее, чем осмысленные имена. Для облегчения

взаимодействия в сети Internet сначала использовались таблицы соответствия числовых адресов именам машин. Эти таблицы сохранились до сих пор и используются многими прикладными программами. Некоторое время даже существовало центральное хранилище соответствий, которое можно было по FTP скачать на свою машину из ftp.internic.net. Это файлы с именем hosts. Если речь идет о системе типа Unix, то этот файл расположен в директории /etc и имеет следующий вид (таблица 12).

Таблица 12 - Пример таблицы имен хостов (файл */etc/hosts*)

IP-адрес	Имя машины	синонимы
127.0.0.1	Localhost	Localhost
144.206.160.32	Polyn	Polyn
144.206.160.40	Apollo	www

Последний столбец в этой таблице является необязательным. Пользователь для обращения к машине может использовать как IP-адрес машины, так и ее имя или синоним (alias). Обращения, приведенные ниже, приводят к одному и тому же результату - инициированию сеанса telnet с машиной Apollo:

telnet 144.206.160.40
или
telnet Apollo
или
telnet www

В локальных сетях файлы hosts используются достаточно успешно до сих пор. Практически все операционные системы поддерживают эту систему соответствия IP-адресов доменным именам.

Однако такой способ присвоения символьных имен был хорош до тех пор, пока Internet был небольшим. По мере роста сети стало затруднительным держать большие списки имен на каждом компьютере. Для того, что бы решить эту проблему, была придумана DNS (Domain Name System).

6.3.1 Принципы организации DNS

Любая DNS является прикладным процессом, который работает над стеком TCP/IP. Таким образом, базовым элементом адресации является IP-адрес, а доменная адресация выполняет роль сервиса. DNS - это информационный сервис Internet, и, следовательно, протоколы его реализующие относятся к протоколам прикладного уровня стандартной модели OSI.

Система доменных адресов строится по иерархическому принципу. Однако иерархия эта не строгая, так как нет единого корня всех доменов Internet. Если более точно, то такой корень в модели DNS есть. Он так и

называется "ROOT". Однако единого администрирования этого корня нет. Администрирование начинается с доменов верхнего, или первого, уровня. В 80-е годы были определены первые домены этого уровня и были рассчитаны на США:

- gov - государственные организации
- mil - военные учреждения
- edu - образовательные учреждения
- com - коммерческие организации
- net - сетевые организации

Позднее, когда сеть перешагнула национальные границы США, появились национальные домены типа:

- uk - Объединенное королевство
- jp - Япония
- au - Австралия
- ch - Чехия
- su - СССР
- ru - Россия

и т.п.

Для СССР также был выделен домен su. После 1991 года, когда республики союза стали суверенными, многие из них получили свои собственные домены: **ua, ru, la, li**, и т.п.

Вслед за доменами первого уровня следуют либо географические домены (kazan.ru, tatarstan.ru), либо организации (kstu.ru). В настоящее время практически любая организация или физическое лицо может получить свой собственный домен второго уровня.

Далее идут домены третьего уровня, например:

efir.kazan.ru

ipm.kstu.ru

Систему доменной адресации можно представлена на рисунке 73.

Служба доменных имен работает как распределенная база, данные которой распределены по DNS-серверам. Сервис DNS строится по схеме "клиент-сервер", где в качестве клиентской части выступает процедура разрешения имен - **resolver,** а в качестве сервера DNS-сервер.

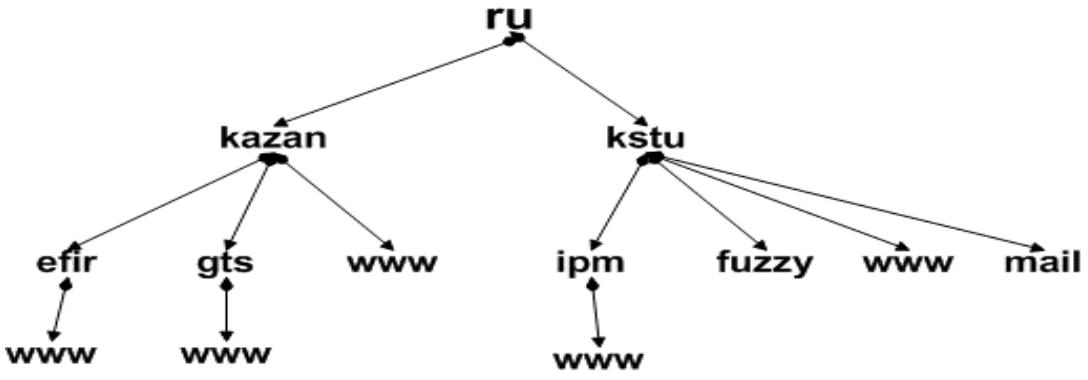


Рисунок 73 – Система доменоной адресации

Например, когда обращаются к серверу ipm.kstu.ru, броузер, используя resolver, поступает следующим образом (рисунок 74):

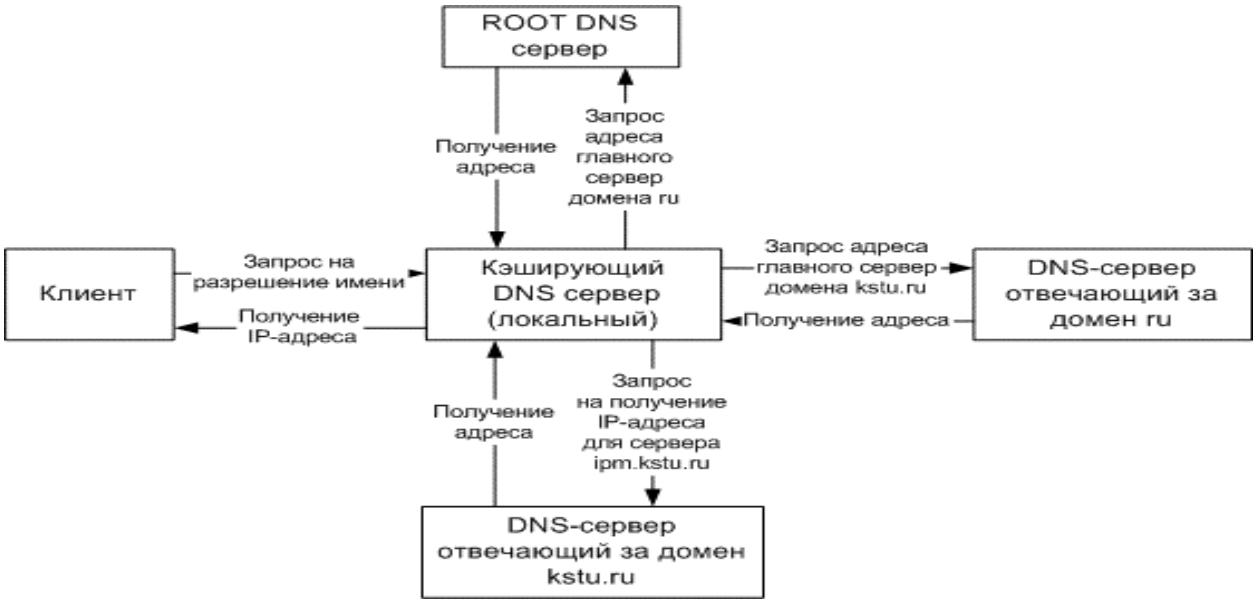


Рисунок 74 – Работа процедуры resolver

1. Ищет запись ipm.kstu.ru в файле hosts, если не находит, то,

2. Посылает запрос на известный DNS-кэширующий сервер (как правило, локальный), если на этом сервере запись не найдена, то,

3. Сервер DNS-кэширующий обращается к DNS-ROOT серверу с запросом адреса DNS сервера отвечающего за домен первого уровня ru, если получает адрес, то,

4. Сервер DNS-кэширующий обращается к DNS серверу, отвечающего за домен первого уровня ru, с запросом адреса DNS сервера отвечающего за домен второго уровня kstu.ru, если получает адрес, то,

5. Сервер DNS-кэширующий посылает запрос на DNS сервер, отвечающий за домен второго уровня kstu.ru, если получает адрес, то,

6. Сервер DNS-кэширующий кэширует адрес и передает клиенту,

7. Клиент обращается по IP адресу - 195.208.44.20.

Первичный - сервер, содержащий полную информацию о зоне.

Вторичный - сервер, содержащий копию полной информации о зоне, полученную с первичного сервера.

Кэширующий - содержит записи которые уже были запрошены

6.3.2 Регистрация доменных имен

Для того, чтобы получить доменный адрес надо отправить заявку в РосНИИРОС (www.ripn.net), который отвечает за делегирование поддоменов в пределах домена ru. В заявке указывается адрес компьютера-сервера доменных имен, почтовый адрес администратора сервера, адрес организации и ряд другой

информации. Разберем эту заявку на примере Международной лаборатории VEGA.

```
domain: vega.ru
descr:  International Agency VEGA
descr:  Kurchatov sq. 1
descr:  115470 Moscow
descr:  Russian Federation
admin-c:        Pavel B Khramtsov
zone-c: Pavel B Khramtsov
tech-c: Pavel B Khramtsov
nserver:        vega-gw.vega.ru
nserver:        ns.relarn.ru
nserver:        polyn.net.kiae.su
dom-net:        194.226.43.0
changed:        paul@kiae.su 961018
source: RIPN

person: Pavel B Khramtsov
address:        International Agency Vega
address:        Kurchatov sq. 1
address:        115470 Moscow
address:        Russian Federation
phone:  +7 095 1969124
fax-no: +7 095 9393670
e-mail: paul@kiae.su
changed:        paul@kiae.su 961018
source: RIPN
```

При заполнении этой заявки следует иметь в виду, что она будет обрабатываться роботом-автоматом. Этот автомат проверяет ее на наличие ошибок заполнения и противоречия с существующей базой данных делегированных доменов. Так как робот не терпит неточностей, то заполнять заявку следует аккуратно. Автомат обрабатывает заявку, выделяет в ней записи для базы данных регистрации доменов. Сами записи состоят из полей, которые идентифицируется в заявке именем поля, после которого ставится символ ":".

В строке описания поля "domain:" указывается имя домена, которое необходимо зарегистрировать. Регистрировать следует как "прямую" зону, так и "обратную.

В строке описания поля "descr:" указывают название и адрес организации, которая запрашивает домен. Так как на одной строке эта информация не может разместиться, то команд descr может быть несколько /26/.

В строке описания поля "admin-c:" указывается персона, которая осуществляет администрирование домена. Поле имеет обязательный формат: <имя> <первая буква отчества> <фамилия>. Таких строк может быть несколько, если лиц отвечающих за администрирование домена больше одного.

Вместо указанного выше формата можно использовать также и идентификатор персоны, если таковой имеется.

Поле "zone-c:" заполняется для той персоны, чей почтовый адрес указан в записи описания зоны. Формат этого поля идентичен формату команды admin-c.

Поле "tech-c:" указывается для технического администратора домена, к которому обращаются в случае экстренных ситуаций. Формат этого поля совпадает с форматом команды admin-c. Во всех полях (admin-c, zone-c и tech-c) имеет смысл указывать координаты одного и того же человека.

В строке описания поля "nserver:" указывают доменные имена серверов зоны. Как правило, таких строк в заявке бывает несколько. Первым указывается primary или главный сервер домена. В нашем случае это vega-gw.vega.ru. На этом сервере хранится база данных домена. Вторым указывается secondary или дублирующий сервер домена. В нашем случае это ns.relarn.ru. Дублирующие сервера призваны повысить надежность работы всей системы доменных имен, поэтому дублирующих серверов может быть несколько. Если существуют другие дублирующие сервера, то указываются и они (сервер polyn.net.kiae.su из нашей заявки также является дублирующим для домена vega.ru). При указании дублирующих серверов первым следует указывать тот, который наиболее надежно обслуживает запросы к домену.

При выборе места размещения дублирующих серверов следует иметь в виду, что сначала проверяется возможность взаимодействия именно с этими серверами. Если хотя бы один из них не откликается на запросы автомата, то заявка отклоняется.

Поле "sub-dom:" описывает поддомены домена. В нашей заявке нет поля sub-net, т.к. в домене vega.ru нет поддоменов. Но если бы нам понадобилось изменять структуру домена, а именно, разбить его на зоны, то тогда следовало бы включить в заявку между строками nserver и строкой dom-net строку:

sub-net: zone1 zone2

В этом случае весь домен vega.ru разбит на два поддомена: zone1 и zone2. Например, машина quest из зоны zone1 будет иметь доменное имя quest.zone1.vega.ru, а машина query из зоны zone2 - query.zone2.vega.ru. Поддомены указываются в команде sub-dom через пробел.

Поле "dom-net:" указывает на список IP-сетей данного домена. В рассматриваемой заявке указана только одна сеть 194.226.43.0, но если у организации, которая заводит свой собственный домен, имеется в наличии несколько сетей, то эти сети можно указать в этом поле, разделяя их пробелами.

Поле "changed:" является обязательным и служит указателем на то лицо, которое последним вносило изменения в заявку. В качестве значения поля после символа ":" указывается почтовый адрес этого лица и, через пробел дата внесения изменения в формате ГГММДД (Г-год, М-месяц, Д-день).

Последняя запись в заявке - это поле source, которое отделяет различные записи во входном потоке данных программы робота. Значение этого поля - RIPN.

Вслед за записью заявки следует запись описания персоны. Именно эта запись позволяет связывать поля admin-c, zone-c, tech-c с информацией о конкретном лице, которая содержится в записи описания персоны.

Запись описания персоны начинается с поля "person:". В данном поле указывается информация о лице (персоне). Обычно, данное поле имеет следующий формат: <имя> <первая буква отчества> <фамилия>.

Поле "address:" вслед за полем "person:". Данное поле состоит из нескольких строк, каждая из которых начинается с имени поля. Первым указывается организация, во второй строке - улица и номер дома, в третьей - почтовый индекс и название города или поселка, в последней строке - название страны. Одним словом - это типичный почтовый адрес.

За адресом следует поле "phone:". В нем указывается номер телефона, по которому можно связаться с указанным лицом. Номер задается с указанием номера страны и кода города. Для Москвы - +7 095 ___ __ __. Телефонов можно указать несколько.

Все выше сказанное для поля "phone:" относится и к полю "fax-no:". В этом поле указывается номер телефакса.

Поле адреса электронной почты - "e-mail:" не является обязательным полем заявки.

В поле "nic-hdl:" указывается персональный номер пользователя, если он у данного пользователя есть. Поле не является обязательным, но, как подчеркивается в инструкции по заполнению заявок "крайне желательно".

Последним полем в описании персоны, как и в заявке на домен, является поле "source:". В заявке можно указать несколько персон, что породит для каждой из них свою собственную запись в базе данных описания доменов. Информация о зоне и лицах, которые ответственны за ведение зоны (и не только о них) можно найти по команде:

/usr/paul>whois -h whois.ripn.net <имя зоны или персоны>

Теперь после описания самой заявки перейдем к описанию ее регистрации. После того как заявка отправлена, следует настроить и запустить сервер в локальном режиме. Администратор РосНИИРОС обычно извещает о том, что у администрации домена ru нет претензий к вашей заявке и разрешает запустить ваш сервер для тестирования.

Заявка отправляется на адрес ru-dns@RIPN.net и попадает на автомат обработки заявок, с которым бесполезно вступать в пререкания и прения по поводу различного рода ошибок и неточностей. Поэтому лучше всего сразу узнать телефон администратора и общаться с ним непосредственно.

При размещении сервера домена также следует позаботиться о том, чтобы существовало от 2-х до 4-х вторичных серверов на случай отказа основного сервера доменных имен.

В РосНИИРОС регистрируют не только "прямую" зону, но также и "обратную". Это две самостоятельные заявки. "Прямая" зона определяет соответствие доменного имени IP-адресу, в то время как "обратная" зона определяет обратное соответствие IP-адреса доменному имени.

6.3.3 Механизм поиска IP-адреса

Очень часто пользователи сообщают администратору системы, что та или иная машина системе не известна, хотя вчера с ней можно было работать. При этом, как правило, называют доменные имена компьютеров. Первое, что следует проверить в этой ситуации - реальную доступность к компьютеру по его IP-адресу, так как если по IP-адресу нельзя "достучаться" до удаленной машины, следует искать ошибки или отказы в работе сервиса доменных имен.

Для этого используется программа named. Так как Resolver, собственно, не является какой-либо программой. Это набор процедур из системной библиотеки, которые позволяют прикладной программе, получать по доменному имени IP-адрес компьютера или по IP-адресу доменное имя. Сами эти процедуры обращаются к системной компоненте resolver, которая ведет диалог с сервером доменных имен и таким образом обслуживает запросы прикладных программ пользователя.

На запросы описанных выше функций в системах Unix отвечает программа named. Идея этой программы проста - обеспечить как разрешение, так называемых, "прямых" запросов, когда по имени ищут адрес, так и "обратных", когда по адресу ищут имя. Управляется named специальной базой данных, которая содержит соответствия между адресами и именами, а также адреса других серверов BIND (Berkeley Internet Name Domain), к которым данный сервер может обращаться в процессе поиска имени или адреса.

Опираясь на схему нерекурсивной процедуры разрешения имени, рассмотрим два способа разрешения запроса на получение IP-адреса по доменному имени.

Первый случай - запрос на получение IP-адреса в рамках зоны ответственности данного местного сервера имен:

1) Прикладная программа через resolver запрашивает IP-адрес по доменному имени у местного сервера.
2) Местный сервер сообщает прикладной программе IP-адрес запрошенного имени.

Несколько примеров, когда появляется запрос на получение IP-адреса по доменному имени:

При входе в режиме удаленного терминала на компьютер polyn.net.kiae.su вводится команда:

/usr/paul>telnet polyn.net.kiae.su

/usr/paul>telnet polyn.net.kiae.su trying

144.206.130.137 ... login: .....

Строчка, в которой указан IP-адрес компьютера polyn.net.kiae.su, показывает, что к этому времени доменное имя было успешно разрешено сервером доменных имен и прикладная программа, в данном случае telnet получила на свой запрос IP-адрес. Таким образом, после ввода команды с консоли и до появления IP-адреса на экране монитора прикладная программа осуществила запрос к серверу доменных имен и получила ответ на него.

Это пример "прямомого" запроса. Но также существуют и “обратные” запросы. В "прямом" запросе прикладная программа запрашивает у сервера доменных имен IP-адрес, сообщая ему доменное имя. При "обратном" запросе прикладная программа запрашивает доменное имя, сообщая серверу доменных имен IP-адрес.

Следует заметить, что скорость разрешения "прямых" и "обратных" запросов в общем случае разная. Все зависит от того, как описаны "прямые" и обратные "зоны" в базах данных серверов доменных имен, обслуживающих домен.

Рассмотрим теперь запрос прикладной программы к серверу доменных имен на получение IP-адреса по доменному имени из домена, который находится в ведении удаленного сервера доменных имен, т.е. сервера отличного от того, домену которого принадлежит компьютер, осуществляющий запрос.

В общем виде такая схема будет выглядеть следующим образом:

1) Прикладная программа обращается к местному серверу доменных имен за IP-адресом, сообщая ему доменное имя.

2) Сервер определяет, что адрес не входит в данный домен и обращается за адресом сервера запрашиваемого домена к корневому серверу доменных имен.

3) Корневой сервер доменных имен сообщает местному серверу доменных имен адрес сервера доменных имен требуемого домена.

4) Местный сервер доменных имен запрашивает удаленный сервер на предмет разрешения запроса своего клиента (прикладной программы)

5) Удаленный сервер сообщает IP-адрес местному серверу.

6) Местный сервер сообщает IP-адрес прикладной программе.

Существует разница между доменом и зоной. Домен - это все множество машин, которые относятся к одному и тому же доменному имени. Однако сам домен разбивается на поддомены или, как их еще называют, зоны. У каждой зоны может быть свой собственный сервер доменных имен. Разбиение домена на зоны и организация сервера для каждой из зон называется делегирование прав управления зоной соответствующему серверу доменных имен, или просто делегированием зоны.

Кроме нерекурсивной процедуры разрешения имен возможна еще и рекурсивная процедура разрешения имен. Ее отличие от описанной выше нерекурсивной процедуры состоит в том, что удаленный сервер сам опрашивает свои серверы зон, а не сообщает их адреса местному серверу доменных имен. Рассмотрим эти два случая более подробно.

6.4 Электронная почта в Internet

Электронная почта - один из важнейших информационных ресурсов Internet. Она является самым массовым средством электронных коммуникаций, предназначена для обмена сообщениями (письмами). Также через почту можно получить доступ к информационным ресурсам других сетей.

Стандартной программой отправки является программа sendmail, которая работает как почтовый курьер, который доставляет обычную почту в отделение связи для дальнейшей рассылки. В Unix-системах sendmail сама является отделением связи. Она сортирует почту и рассылает ее адресатам /27/.

Электронная почта работает по принципу “клиент”-“сервер”. Клиент (MS Outlook, The bat ...) готовит ("упаковывает") и посылает серверу (почтовое отделение) сообщения, принимает и просматривает сообщения. Сервер электронной почты (Sendmail, MS Exchange ...) обрабатывает сообщения (сортирует) и отправляет локальному адресату или удаленному серверу (почтовому отделению) (рисунок 75).

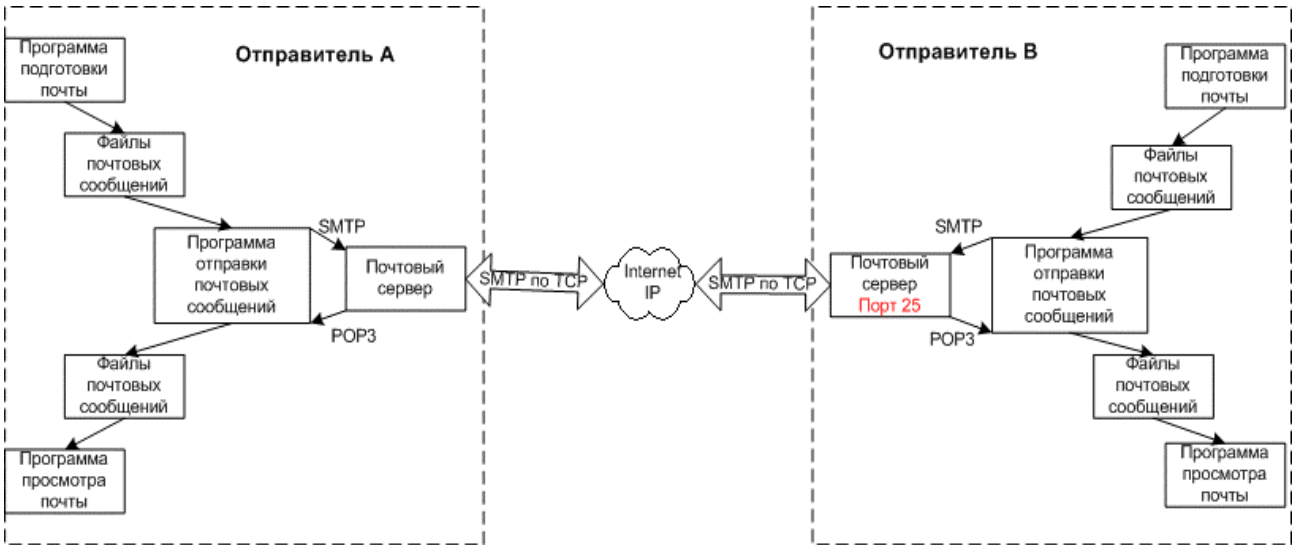


Рисунок 75 - Структура взаимодействия участников почтового обмена

Основные протоколы:

– SMTP (Simple Mail Transfer Protocol) - простой протокол передачи почты, используется для **отправки** почты, как клиентом на сервер, так и сервером на другой сервер.

– POP3 (Post Office Protocol) - используется для **приема** почты клиентом с сервера.

– UUCP (Unix-Unix-CoPy) - используется для **отправки и приема** почты, как клиентом на (с) сервер(а), так и сервером на другой сервер. В данное время почти не используется, поэтому рассматривать не будем.

SMTP является протоколом прикладного уровня и использует транспортный протокол TCP. Совместно с этим протоколом используется и UUCP протокол. Разница между SMTP и UUCP заключается в том, что при использовании первого протокола sendmail пытается найти машину-получателя почты и установить с ней взаимодействие в режиме on-line для того, чтобы передать почту в ее почтовый ящик, почта достигает почтового ящика получателя за считанные минуты и время получения сообщения зависит только от того, как часто получатель просматривает свой почтовый ящик. При использовании же UUCP почта передается по принципу "stop-go", т.е.

сообщение передается по цепочке почтовых серверов от одной машины к другой пока не достигнет машины-получателя или не будет отвергнуто по причине отсутствия абонента-получателя. С одной стороны, UUCP позволяет доставлять почту по плохим телефонным каналам, а с другой стороны можно получить возврат сообщения через сутки после его отправки из-за того, что допущена ошибка в имени пользователя.

Основой любой почтовой службы является система адресов. В Internet принята система адресов, которая базируется на доменном адресе машины, подключенной к сети. Например, для пользователя paul машины с адресом polyn.net.kiae.su почтовый адрес будет выглядеть как:

paul@polyn.net.kiae.su.

Таким образом, адрес состоит из двух частей: идентификатора пользователя, который записывается перед знаком "коммерческого @", и доменного адреса машины, который записывается после знака "@".

6.4.1 Протокол SMTP

SMTP - Simple Mail Transfer Protocol), использует порт по умолчанию - 25. Основной недостаток протокола, это отсутствие аутентификации и "докачки" (как в FTP, HTTP) сообщений, т.е. если посылается большое письмо (10Мбайт), то в случае разрыва соединения сообщение придется передавать заново. Поэтому большие письма необходимо резать на части.

Модель протокола: клиент инициирует соединение с сервером; клиент посылает запросы на обслуживание; сервер отвечает на эти запросы.

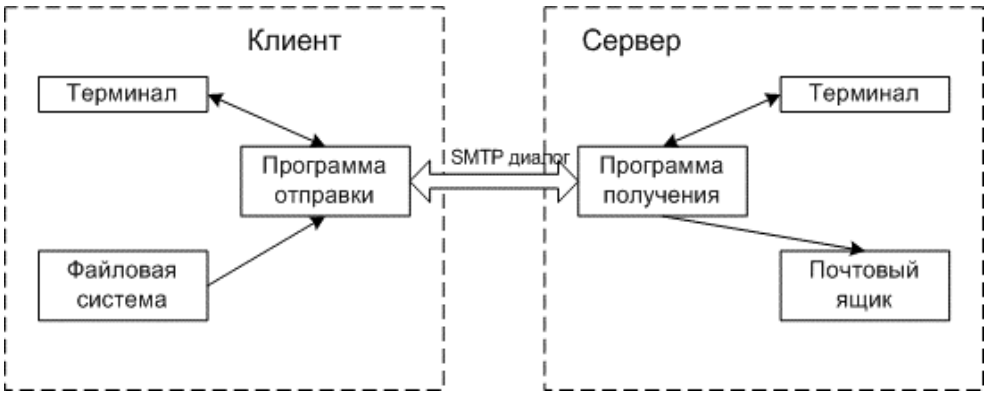


Рисунок 76 - Модель протокола SMTP

6.4.2 Протокол POP

Post Office Protocol (POP) - протокол доставки почты пользователю из его почтового ящика почтового сервера POP. Когда почта пришла на сервер (по SMTP), она раскладывается по почтовым ящикам. Чтобы забрать почту из ящика нужен POP.

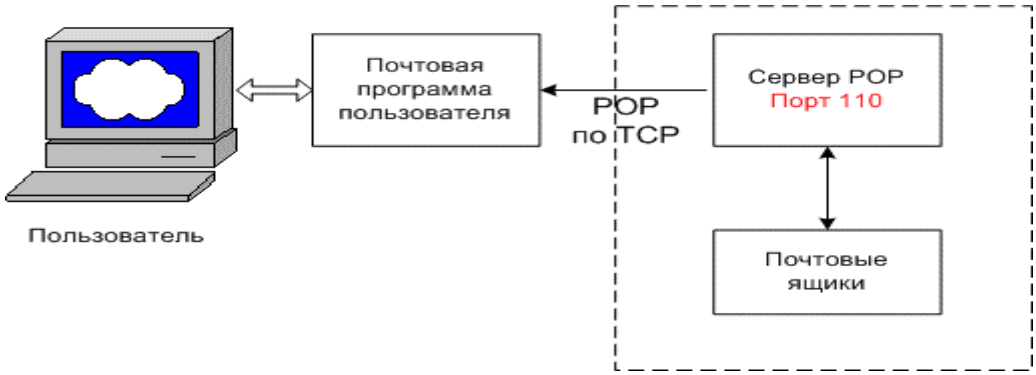


Рисунок 77 - Модель протокола POP

В протоколе POP3 оговорены три стадии процесса получения почты:

– авторизация;
– транзакция;
– обновление (завершение транзакции).

После того как сервер и клиент POP3 установили соединение, начинается стадия авторизации. На стадии авторизации клиент идентифицирует себя для сервера. Если авторизация прошла успешно, сервер открывает почтовый ящик клиента и начинается стадия транзакции. В ней клиент либо запрашивает у сервера информацию (например, список почтовых сообщений), либо просит его совершить определенное действие (например, выдать почтовое сообщение). Наконец, на стадии обновления сеанс связи заканчивается.

Ответ сервера может иметь либо положительный, либо отрицательный ответ.

6.4.4 Формат представления почтовых сообщений MIME

MIME означает "Multipurpose Internet Mail Extensions" (Многоцелевые расширения почтового стандарта Internet). Этот стандарт описывает, как пересылать по электронной почте исполняемые, графические, мультимедийные, смешаные данные. Типичные применения MIME - пересылка графических изображений, аудио, документов Word, программ и даже просто текстовых файлов, то есть, когда важно, чтобы входе пересылки не производилось никаких преобразований над данными. MIME также позволяет размечать письмо на части различных типов так, чтобы получатель (почтовая программа) мог определить, что делать с каждой из частей письма.

Так как файлы могут быть разными (.gif, .doc, .pdf ...), броузер должен понимать, что с ними делать. Эту проблему решает стандарт "MIME - типы". Он сообщает клиенту, какой тип файлов получен, например:Content-type: image/gif (графика GIF) Content-type: image/jpeg (графика JPG)

Броузеры используют MIME-типы в своих HTTP-заголовках для того, чтобы сообщить, в каких форматах они предпочитают принимать данные (если сервер может выдать файл в разных форматах). Серверы используют MIME-типы в HTTP-заголовках Content-Type, чтобы сообщить клиенту о том, в каком формате передается прилагаемое содержимое: то ли это HTML, который нужно

форматировать, то ли это GIF или JPEG, требующий визуализации, то ли это данные в формате PDF, для которого нужно открывать внешнюю программу просмотра или использовать дополнительное приложение.

Стандарт MIME предназначен для описания тела почтового сообщения Internet. Предшественником MIME является Стандарт почтового сообщения ARPA (RFC-822). Стандарт RFC-822 был разработан для обмена текстовыми сообщениями. С момента опубликования стандарта возможности аппаратных средств и телекоммуникаций ушли далеко вперед и стало ясно, что многие типы информации, которые широко используются в сети, невозможно передать по почте без специальных преобразований. Так в тело сообщения нельзя включить графику, аудио, видео и другие типы информации.

6.5 Удаленный доступ к ресурсам сети. Протокол Telnet

Telnet - это одна из самых старых информационных технологий Internet.

Назначение Telnet-протокола - дать общее описание взаимодействия терминального устройства и терминал-ориентированного процесса. При этом этот протокол может быть использован и для организации взаимодействий "терминал-терминал" (связь) и "процесс-процесс" (распределенные вычисления). Стандартным портом TCP для telnet является порт 23.

Telnet строится как протокол приложения над транспортным протоколом TCP. В основу telnet положены три фундаментальные идеи:

– концепция сетевого виртуального терминала (Network Virtual Terminal) или NVT;

– принцип договорных опций (согласование параметров взаимодействия);

– симметрия связи "терминал-процесс".

1) В протоколе Telnet NVT определен как "двунаправленное символьное устройство, состоящее из принтера и клавиатуры". Принтер предназначен для отображения приходящей по сети информации, а клавиатура - для ввода данных, передаваемых по сети. NVT предполагается буферизованным устройством.

Это означает, что данные, вводимые с клавиатуры, не посылаются сразу по сети, а собираются в пакеты, которые отправляются либо по мере заполнения буфера, либо по специальной команде. Такая организация NVT призвана с одной стороны, минимизировать сетевой трафик, а с другой обеспечить совместимость с реальными буферизованными терминалами.

NVT - это стандартное описание наиболее широко используемых возможностей реальных физических терминальных устройств. NVT позволяет описать и преобразовать в стандартную форму способы отображения и ввода информации. Терминальная программа ("user") и процесс ("server"), работающий с ней, преобразовывают характеристики физических устройств в спецификацию NVT, что позволяет, с одной стороны, унифицировать характеристики физических устройств, а с другой обеспечить принцип

совместимости устройств с разными возможностями. Характеристики диалога диктуются устройством с меньшими возможностями.

Если взаимодействие осуществляется по принципу "терминал-терминал" или "процесс-процесс", то "user" - это сторона, инициирующая соединение, а "server" - пассивная сторона.

2) Принцип договорных опций или команд позволяет согласовать возможности представления информации на терминальных устройствах. NVT - это минимально необходимый набор параметров, который позволяет работать по telnet даже самым допотопным устройствам, реальные современные устройства обладают гораздо большими возможностями представления информации. Принцип договорных команд позволяет использовать эти возможности. Симметрия взаимодействия по протоколу telnet позволяет в течение одной сессии программе-"user" и программе-"server" меняться местами. Это принципиально отличает взаимодействие в рамках telnet от традиционной схемы "клиент-сервер".

3)Симметрия взаимодействия тесно связана с процессом согласования формы обмена данными между участниками telnet-соединения. Когда речь идет о работе на удаленной машине в режиме терминала, то возможности ввода и отображения информации определяются только конкретным физическим терминалом и договорной процесс сводится к заказу терминальной программой характеристик этого терминала. Гораздо сложнее обстоит дело, когда речь идет об обмене информацией между двумя терминальными программами в режиме "терминал-терминал". В этом случае каждая из сторон может выступать инициатором изменения принципов представления информации, и здесь проявляется еще одна особенность протокола telnet. Протокол не использует принцип "запрос-подтверждение", а применяет принцип "прямого действия". Это значит, что если терминальная программа хочет расширить возможности представления информации, то она делает это, если в ответ она получает информацию в новом представлении, то это означает, что попытка удалась, в противном случае происходит возврат к стандарту NVT.

6.6 Служба архивов FTP

Технология FTP была разработана в рамках проекта ARPA и предназначена для обмена большими объемами информации между машинами с различной архитектурой. Главным в проекте было обеспечение надежной передачи и поэтому с современной точки зрения FTP кажется перегруженным излишними редко используемыми возможностями. Стержень технологии составляет FTP-протокол.

FTP-архивы являются одним из основных информационных ресурсов Internet. Фактически, это распределенное хранилище текстов, программ, фильмов, фотографий, аудио записей и прочей информации, хранящейся в виде файлов на различных компьютерах во всем мире.

Типы информационных ресурсов. Информация в FTP-архивах разделена на три категории:

– защищенная информация, режим доступа к которой определяется ее владельцами и разрешается по специальному соглашению с потребителем. К этому виду ресурсов относятся коммерческие архивы, закрытые некоммерческие ресурсы, частная некоммерческая информация (частные благотворительные фонды);

– информационные ресурсы ограниченного использования. В данный класс могут входить ресурсы ограниченного времени использования (текущая версия Netscape перестанет работать в июне, если только кто-то не сломает защиту) или ограниченного времени действия, т.е. пользователь может использовать текущую версию, но никто не будет оказывать ему поддержку;

– свободно распространяемые информационные ресурсы. К этим ресурсам относится все, что можно свободно получить по сети без специальной регистрации. Это может быть документация, программы или что-либо еще. Следует отметить, что свободно распространяемое программное обеспечение не имеет сертификата качества, но, как правило, его разработчики открыты для обмена опытом.

Из выше перечисленных ресурсов наиболее интересными, по понятным причинам, являются две последних категории, которые, как правило, оформлены в виде FTP-архивов.

Служба FTP (от протокола - File Transfer Protocol) - предназначена для обмена файлами и FTP служба построена по "клиент-сервер".

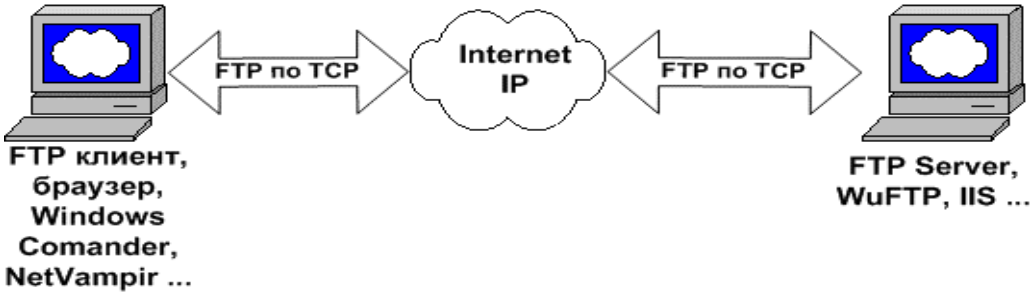


Рисунок 78 - Работа FTP на пользовательском уровне

Клиент (броузер,Windows Commander, NetVampir ...) посылает запросы серверу и принимает файлы.

Сервер HTTP (Apache, IIS ...) обрабатывает запросы клиента на получение файла.

Служба FTP базируется на двух стандартах:

– URL (Universal Resource Locator) - универсальный способ адресации ресурсов в сети;

– FTP (File Transfer Protocol) - протокол передачи файлов.

6.6.1 Протокол FTP

File Transfer Protocol - уровня приложений. Используется службой FTP для передачи файлов. FTP отличается от других приложений тем, что он использует два TCP соединения для передачи файла.

1 Управляющее соединение - соединение для посылки команд серверу и получение ответов от него. Для канала управления используется протокол Telnet.

2. Соединение данных - соединение для передачи файлов (рисунок 79).

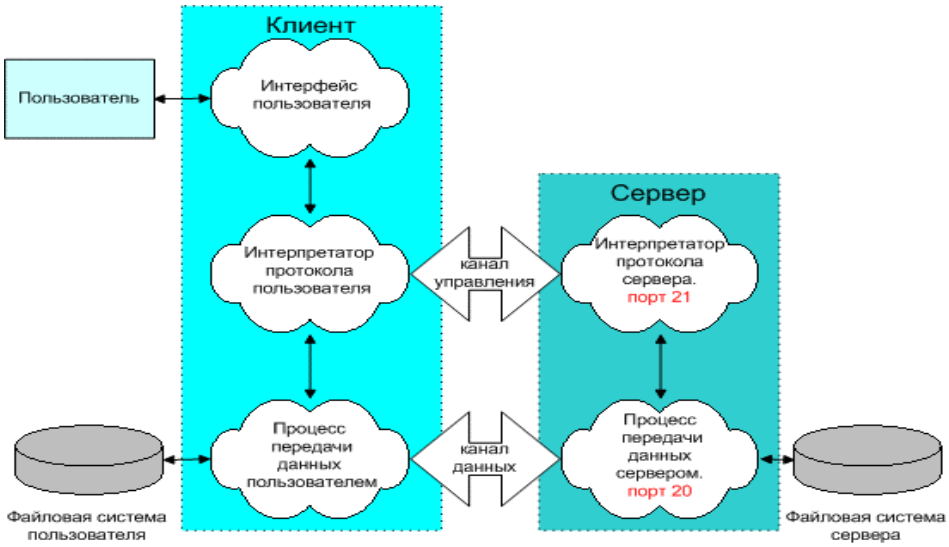


Рисунок 79 – Работа протокола FTP

В FTP соединение инициируется интерпретатором протокола пользователя. Управление обменом осуществляется по каналу управления в стандарте протокола Telnet. Команды FTP генерируются интерпретатором протокола пользователя и передаются на сервер. Ответы сервера отправляются пользователю также по каналу управления. В общем случае пользователь имеет возможность установить контакт с интерпретатором протокола сервера и отличными от интерпретатора пользователя средствами.

Команды FTP определяют параметры канала передачи данных и самого процесса передачи. Они также определяют и характер работы с удаленной и локальной файловыми системами. Сессия управления инициализирует канал передачи данных. При организации канала передачи данных последовательность действий другая, отличная от организации канала управления. В этом случае сервер инициирует обмен данными в соответствии с согласованными в сессии управления параметрами.

Канал данных устанавливается для того же host'a, что и канал управления, через который ведется настройка канала данных. Канал данных может быть использован как для приема, так и для передачи данных.

Возможна ситуация, когда данные могут передаваться на третью машину. В этом случае пользователь организует канал управления с двумя серверами и

организует прямой канал данных между ними. Команды управления идут через пользователя, а данные напрямую между серверами.

Канал управления должен быть открыт при передаче данных между машинами. В случае его закрытия передача данных прекращается.

Протокол FTP определяет запрос-ответный способ взаимодействия между программой-клиентом и программой-сервером.

Работа FTP на пользовательском уровне содержит несколько этапов:

1. Идентификация (ввод имени и пароля).
2. Выбор каталога.
3. Определение режима обмена (поблочный, поточный, ascii или двоичный).
4. Выполнение команд обмена (get, mget, dir, mdel, mput или put).
5. Завершение процедуры (quit или close).

В старых версиях для передачи данных использовался только 20-й порт (активный режим), в современных версиях FTP-серверов порт для канала данных может назначается сервером из нестандартных (N > 1024) портов (пассивный режим).

Различие работы пассивного режима и активного. Активный режим.

Действия сервера и клиента:

1. Клиент устанавливает связь и посылает запрос на 21 порт сервера с порта N (N>1024)
2. Сервер посылает ответ на порт N (N>1024) клиента
3. Сервер устанавливает связь для передачи данных по порту 20 на порт клиента N+1

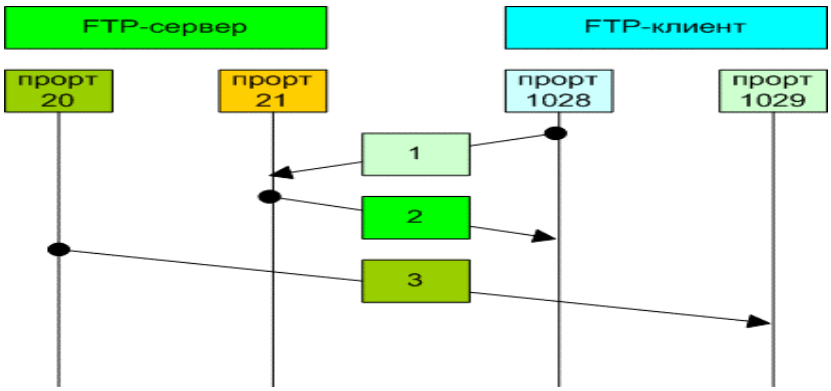


Рисунок 80 - Активный режим работы FTP

Пассивный режим. Действия сервера и клиента:

1. Клиент устанавливает связь и посылает запрос (сообщает, что надо работать в пассивном режиме) на 21 порт сервера с порта N (N>1024)
2. Сервер посылает ответ и сообщает номер порта для канала данных P (P>1024) на порт N (N>1024) клиента

3. Клиент устанавливает связь для передачи данных по порту N+1 на порт сервера P (P>1024)

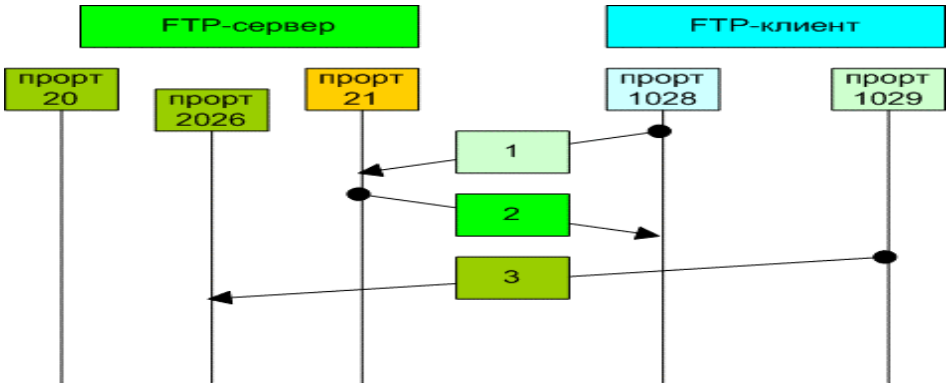


Рисунок 81 - Пассивный режим работы FTP

Активный FTP выгоден для FTP-сервера, но вреден для стороны клиента. FTP сервер пытается соединиться со случайными высокими (по номеру) портами на клиенте, такое соединение, наверняка, будет блокировано на стороне клиента.

Пассивный FTP выгоден для клиента, но вреден для FTP-сервера. Клиент будет делать оба соединения к серверу, но одно из них будет к случайному высокому порту, такое соединение, наверняка, будет блокировано на стороне сервера.

6.7 Универсальный идентификатор ресурсов URI

URI (Uniform Resource Identifier, Универсальный идентификатор ресурса) - компактная строка символов для идентификации абстракного или физического ресурса. Под ресурсом понимается любой объект, принадлежащий некоторому пространству.

Необходимость в URI была понятна разработчикам WWW с момента зарождения системы, т.к. предполагалось объединение в единую информационную среду средств, использующих различные способы идентификации информационных ресурсов. Была разработана спецификация, которая включала в себя обращения к FTP, Gopher, WAIS, Usenet, E-mail, Prospero, Telnet, X.500 и, конечно, HTTP (WWW). В итоге была разработана универсальная спецификация, которая позволяет расширять список адресуемых ресурсов за счет появления новых схем.

Место применения URI - гипертекстовые ссылки, которые записываются в тегах <A HREF=URI> и <LINK HREF=URI>. Встраиваемые графические объекты также адресуются по спецификации URI в тегах <IMG SRC=URI> и <FIG SRC=URI>. Реализация URI для WWW называется URL (Uniform Resource Locator). Точнее, URL - это реализация схемы URI, отображенная на алгоритм доступа к ресурсам по сетевым протоколам. Существует еще и URN (Uniform Resource Name), которое отображает URI в пространство имен на сети.

Появление URN связано с желанием адресовать части почтового сообщения MIME.

Принципы построения адреса WWW. В основу URI были заложены следующие принципы:

– *Расширяемость* - новые адресные схемы должны легко вписываться в существующий синтаксис URI.

– *Полнота* - по возможности, любая из существовавших схем должна описываться посредством URI.

– *Читаемость* - адрес должен был быть легко читаем пользователем, что вообще характерно для технологии WWW - документы вместе с ссылками могут разрабатываться в обычном текстовом редакторе.

Прежде, чем рассмотреть различные схемы представления адресов приведем пример простого адреса URI:

http://polyn.net.kiae.su/polyn/index.html

Перед двоеточием стоит идентификатор схемы адреса - "http". Это имя отделено двоеточием от остатка URI, который называется "путь". В данном случае путь состоит из доменного адреса машины, на которой установлен сервер HTTP и пути от корня дерева сервера к файлу "index.html".

Кроме представленной выше полной записи URI, существует упрощенная. Она предполагает, что к моменту ее использования многие параметры адреса ресурса уже определены (протокол, адрес машины в сети, некоторые элементы пути). При таких предположениях автор гипертекстовых страниц может указывать только относительный адрес ресурса, т.е. адрес относительно определенных базовых ресурсов /27/.

Некоторые подмножества URI:

URL (Uniform Resource Locator, Универсальный указатель ресурса), - подмножество схем URI, который идентифицирует ресурс по способу доступа к нему (например, его "местонахождению в сети") вместо того, чтобы идентифицировать его по названию или другим атрибутам этого ресурса.

URL - Uniform Resource Locators явно описывает, как добраться до объекта.

Синтаксис:

<scheme>:<scheme-specific-part>

где:

scheme = "http" | "ftp" | "gopher" | "mailto" | "news" | "telnet" | "file" | "man" | "info" | "whatis" | "ldap" | "wais" | ... - имя схемы

scheme-specific-part - зависит от схемы

В scheme-specific-part можно использовать шестнадцатеричные значения в виде: %5f. Обязательно должны кодироваться непечатные октеты: 00-1F, 7F, 80-FF.

Примеры URL:

http://www.ipm.kstu.ru/index.php

ftp://www.ipm.kstu.ru/

В HTML записывается так:

URN (Uniform Resource Name, Универсальное имя ресурса) - частная URI-схема "urn:" с подмножеством "пространства имен", который должен быть уникальным и неизменным даже в том случае, когда ресурс уже не существует или недоступен.

Предполагается что, например броузер, знает, где искать этот ресурс.

Синтаксис:

urn:namespace: data1.data2,more-data, где namespace (пространство имен) определяет, каким образом используются данные, указанные после второго ":".

Пример URN:

urn:ISBN: 0-395-36341-6

ISBN - тематический классификатор для издательств

0-395-36341-6 - конкретный номер тематики книги или журнала

При получении URN клиентская программа обращается к ISBN (каталогу "тематический классификатор для издательств" в Интернете). И получает расшифровку номера тематики "0-395-36341-6" (например: "квантовая химия").

URN принят сравнительно недавно, в текущие версии HTML не включен и службы каталогов пока не развиты, поэтому URN не так широко распространен как URL.

6.7.1 Схемы адресации ресурсов Internet

Существует 8 схем адресации ресурсов Internet. В схеме указывается ее идентификатор, адрес машины, TCP-порт, путь в директории сервера, переменные и их значения, метка.

Схема HTTP. Это основная схема для WWW. В схеме указывается ее идентификатор, адрес машины, TCP-порт, путь в директории сервера, поисковый критерий и метка.

Синтаксис:

http://[<user>[:<password]>@]<host>[:<port>][/[<url-path>][?<query>]]

http - название схемы

user - имя пользователя

password - пароль пользователя

host - имя хоста

port - номер порта

url-path - путь к файлу и сам файл

query (<имя-поля>=<значение>{&<имя-поля>=<значение>) - строка запроса

По умолчанию, port=80.

Приведем несколько примеров URI для схемы HTTP:

http://polyn.net.kiae.su/polyn/manifest.html

Это наиболее распространенный вид URI, применяемый в документах WWW. Вслед за именем схемы (http) следует путь, состоящий из доменного адреса машины и полного адреса HTML-документа в дереве сервера HTTP.

В качестве адреса машины допустимо использование и IP-адреса:

http://144.206.160.40/risk/risk.html

Если сервер протокола HTTP запущен на другой, отличный от 80 порт TCP, то это отражается в адресе:

http://144.206.130.137:8080/altai/index.html

При указании адреса ресурса возможна ссылка на точку внутри файла HTML. Для этого вслед за именем документа может быть указана метка внутри документа:

http://polyn.net.kiae.su/altai/volume4.html#first

Схема FTP. Данная схема позволяет адресовать файловые архивы FTP из программ-клиентов World Wide Web. При этом программа должна поддерживать протокол FTP. В данной схеме возможно указание не только имени схемы, адреса FTP-архива, но и идентификатора пользователя и даже его пароля.

Синтаксис:

ftp://[<user>[:<password]>@]<host>[:<port>][/<url-path>]

ftp - название схемы

user - имя пользователя

password - пароль пользователя

host - имя хоста

port - номер порта

url-path - путь к файлу и сам файл

По умолчанию, port=21, user=anonymous, password=email-адрес.

Наиболее часто данная схема используется для доступа к публичным архивам FTP:

ftp://polyn.net.kiae.su/pub/0index.txt

В данном случае записана ссылка на архив "polyn.net.kiae.su" с идентификатором "anonymous" или "ftp" (анонимный доступ). Если есть необходимость указать идентификатор пользователя и его пароль, то можно это сделать перед адресом машины:

ftp://nobody:password@polyn.net.kiae.su/users/local/pub

В данном случае эти параметры отделены от адреса машины символом "@", а друг от друга двоеточием.

Схема TELNET. По этой схеме осуществляется доступ к ресурсу в режиме удаленного терминала. Обычно клиент вызывает дополнительную программу для работы по протоколу telnet. При использовании этой схемы необходимо указывать идентификатор пользователя, допускается использование пароля.

Синтаксис:

telnet://[<user>[:<password]>@]<host>[:<port>]/

telnet - название схемы

user - имя пользователя

password - пароль пользователя

host - имя хоста

port - номер порта

По умолчанию, port=23.

Пример:
telnet://name:password@ipm.kstu.ru

Реально, доступ осуществляется к публичным ресурсам, и идентификатор и пароль являются общеизвестными, например, их можно узнать в базах данных Hytelnet.

telnet://guest:password@apollo.polyn.kiae.su

Из приведенных выше примеров видно, что спецификация адресов ресурсов URI является довольно общей и позволяет проидентифицировать практически любой ресурс Internet. При этом число ресурсов может расширяться за счет создания новых схем.

6.8 Служба WWW

Служба WWW (World Wide Web) - предназначена для обмена гипертекстовой информацией, построена по схеме "клиент-сервер".

Броузер (Internet Explorer, Opera ...) является мультипротокольным клиентом и интерпретатором HTML. И как типичный интерпретатор, клиент в зависимости от команд (тегов) выполняет различные функции. В круг этих функций входит не только размещение текста на экране, но обмен информацией с сервером по мере анализа полученного HTML-текста, что наиболее наглядно происходит при отображении встроенных в текст графических образов.

Сервер HTTP (Apache, IIS ...) обрабатывает запросы клиента на получение файла.

В начале служба WWW базировалась на трех стандартах:

– HTML (HyperText Markup Lan-guage) - язык гипертекстовой разметки документов;

– URL (Universal Resource Locator) - универсальный способ адресации ресурсов в сети;

– HTTP (HyperText Transfer Protocol) - протокол обмена гипертекстовой информацией.

Позже добавили CGI (Common Gateway Interface) - универсальный интерфейс шлюзов. Создан для взаимодействия HTTP - сервера с другими программами установленными на сервере (например, СУБД).

6.8.1 Схема работы WWW сервера

WWW сервер - это такая часть глобальной или внутрикорпоративной сети, которая дает возможность пользователям сети получать доступ к гипертекстовым документам, расположенным на данном сервере. Для взаимодействия с WWW сервером пользователь сети должен использовать специализированное программное обеспечение - броузер (от англ. browser) - программа просмотра.

Рассмотрим более схему работы WWW-сервера:

1. Пользователь сети запускает броузер**,** в функции которого входит:

- установление связи с сервером;
- получение требуемого документа;
- отображение полученного документа;
- реагирование на действия пользователя - доступ к новому документу.

После запуска броузер по команде пользователя или автоматически устанавливает связь с заданным WWW - сервером и передает ему запросполучение заданного документа.

2. WWW сервер ищет запрашиваемый документ и возвращает результаты броузеру.

3. Броузер, получив документ, отображает его пользователю и ожидает его реакции. Возможные варианты:

- ввод адреса нового документа;
- печать, поиск, другие операции над текущим документом;
- активизация (нажатие) специальных зон полученного документа, называемых связями (link) и ассоциированными с адресом нового документа. В первом и третьем случае происходит обращение за новым документом.

6.8.2 Архитектура построения системы

От описания основных компонентов перейдем к архитектуре взаимодействия программного обеспечения в системе World Wide Web. WWW построена по хорошо известной схеме "клиент-сервер". На рисунке 82 показано, как разделены функции в этой схеме.

Рисунок 82 - Архитектура WWW-технологии

Программа-клиент выполняет функции интерфейса пользователя и обеспечивает доступ практически ко всем информационным ресурсам Internet. Фактически, клиент - это интерпретатор HTML. При анализе URL-

спецификации или по командам сервера клиент запускает дополнительные внешние программы для работы с документами в форматах, отличных от HTML, например GIF, JPEG и т.п.

Другую часть программного комплекса WWW составляет сервер протокола HTTP, базы данных документов в формате HTML, управляемые сервером, и программное обеспечение, разработанное в стандарте спецификации CGI. Появился очень неплохой сервер для MS-Windows и Apachie-сервер для Unix- платформ. Существуют и другие, но два последних можно выделить из соображений доступности использования.

База данных HTML-документов - это часть файловой системы, которая содержит текстовые файлы в формате HTML и связанные с ними графику и другие ресурсы.

Прикладное программное обеспечение, работающее с сервером, можно разделить на программы-шлюзы и прочие. Шлюзы - это программы, обеспечивающие взаимодействие сервера с серверами других протоколов. Прочие программы - это программы, принимающие данные от сервера и выполняющие какие-либо действия: получение текущей даты, реализацию графических ссылок, доступ к локальным базам данных или просто расчеты.

Все, что было сказано до этого момента, можно отнести к классической схеме World Wide Web. В настоящее время следует говорить об изменении общей архитектуры.

Как видно из рисунка 83, к середине 1996 года произошли некоторые изменения в архитектуре сервиса World Wide Web. Произошел возврат к модульной структуре сервера World Wide Web. Этот возврат был реализован в виде спецификации API. API - это спецификация разработки прикладных модулей, которые встраиваются в сервер.

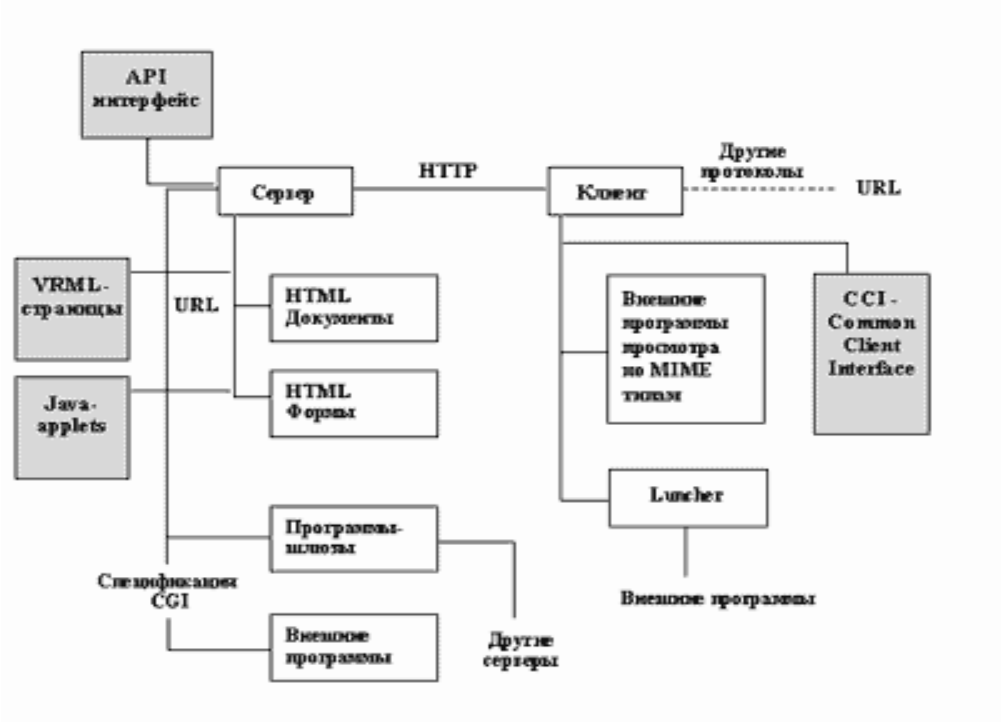


Рисунок 83 - Архитектура World Wide Web к середине 1996 года

Применение во всех серверах многопотоковой технологии выполнения подзадач делает такой способ расширения возможностей сервера более экономичным с точки зрения ресурсов вычислительной установки, чем разработка CGI-скриптов.

В дополнение к HTML активно стал применяться еще один язык разметки - VRML (Virtual Reality Modeling Language). В данном случае речь идет об описании трехмерных сцен и возможности "бродить" по этим мирам.

Java-applet'ы - это мобильные коды Java, ссылки на которые вмонтированы в тело документа. При доступе к такому документу программа просмотра пользователя предварительно анализирует документ на предмет наличия в нем такого типа ссылок, и, если они существуют, то подкачивает мобильные коды в свою память.

Как видно из рисунка, изменения коснулись и клиентской части технологии. В настоящее время происходит постепенный переход от простой классической архитектуры клиент-сервер к архитектуре с сервером приложений, в роли которого выступает программа-клиент - CCI (Common Client Interface).

6.9 Протокол обмена гипертекстовой информацией

HTTP - это протокол прикладного уровня, разработанный для обмена гипертекстовой информацией в сети Internet, используется Word Wide Web с 1990 года.

Реальная информационная система требует гораздо большего количества функций, чем просто поиск. HTTP позволяет реализовать в рамках обмена данными набор методов доступа, базирующихся на спецификации универсального идентификатора ресурсов (URI), применяемого в форме универсального локатора ресурсов (URL) или универсального имени ресурса (URN). Сообщения по сети при использовании протокола HTTP передаются в формате, схожим с форматом почтового сообщения Internet или с форматом сообщений MIME. HTTP используется для взаимодействия программ-клиентов с программами-шлюзами, разрешающими доступ к ресурсам электронной почты Internet (SMTP), спискам новостей (NNTP), файловым архивам (FTP), системам Gopher и WAIS. Протокол разработан для доступа к этим ресурсам посредством промежуточных программ-серверов (proxy), которые позволяют передавать информацию между различными информационными службами без потерь. Протокол реализует принцип "запрос/ответ". Запрашивающая программа - клиент - инициирует взаимодействие с отвечающей программой - сервером, и посылает запрос, включающий в себя метод доступа, адрес URI, версию протокола, информацию клиента, и, возможно, тело сообщения клиента. Сервер отвечает строкой состояния, включающей версию протокола и код возврата. Данное сообщение содержит информацию сервера, метаинформацию и тело сообщения.

При работе в Internet для обслуживания HTTP-запросов используется 80 порт TCP/IP. Практика использования протокола такова, что клиент

устанавливает соединение и ждет ответа сервера. После отправки ответа сервер инициирует разрыв соединения. Таким образом, при передаче сложных гипертекстовых страниц соединение может устанавливаться несколько раз. Остановимся более подробно на механизме взаимодействия и форме передаваемой информации.

Форма запроса клиента. Программа-клиент посылает после установления соединения запрос серверу. Этот запрос может быть в двух формах: в форме полного запроса и в форме простого запроса. Простой запрос содержит метод доступа и запрос ресурса. Например:

GET http://polyn.net.kiae.su/

В этой записи слово GET обозначает метод доступа GET, а http://polyn.net.kiae.su/ - это запрос ресурса.

Полная форма содержит тип протокола доступа, адрес сервера ресурса, и адрес ресурса на сервере (рисунок 84).

Рисунок 84 - Полный адрес ресурса

Общий вид полного запроса выглядит так:

<Полный запрос> := <Строка Запроса> (<Общий заголовок>|<Заголовок запроса>|<Заголовок обозначения ресурса>)<символ новой строки>[<тело ресурса>]

Квадратные скобки здесь обозначают необязательные элементы заголовка. Строка запроса - это, практически, простой запрос ресурса. Отличие состоит в том, что в строке запроса можно указывать различные методы доступа и за запросом ресурса следует указывать версию протокола. Например, для вызова внешней программы можно использовать следующую строку запроса:

POST http://polyn.net.kiae.su/cgi-bin/test HTTP/1.0

В данном случае используется метод POST и протокол версии 1.0.

В обеих формах запроса важное место занимает форма запроса ресурса, которая кодируется в соответствии со спецификацией URI. Применительно к World Wide Web эта спецификация получила название URL. При обращении к серверу можно использовать как полную форму URL, так и упрощенную.

Методы доступа. В настоящее время в практике World Wide Web реально используются только три метода доступа: POST, GET, HEAD.

GET - метод, позволяющий получить данные, заданные в форме URI в запросе ресурса. Если ссылаются на программу, то возвращается результат выполнения этой программы, но не текст программы. Дополнительные данные, которые надо передать для обработки, кодируются в запрос ресурса. При

использовании метода GET в поле тела ресурса возвращается собственно затребованная информация (текст HTML-документа, например).

HEAD - метод, аналогичный GET, но не возвращает тела ресурса. Используется для получения информации о ресурсе и для тестирования гипертекстовых ссылок.

POST - этот метод разработан для передачи большого объема информации на сервер. Им пользуются для аннотирования существующих ресурсов, посылки почтовых сообщений, работы с формами интерфейсов к внешним базам данных и внешним исполняемым программам. В отличие от GET и HEAD, в POST передается тело ресурса, которое и является информацией из поля форм или других источников ввода.

Ответ сервера. Ответ сервера может быть, как и запрос, упрощенным или полным. При упрощенном ответе сервер возвращает только тело ресурса (например, текст HTML-документа). При полном ответе клиенту возвращается строка состояния, общий заголовок, заголовок ответа, заголовок ресурса и тело ресурса. Полный ответ представляется следующим образом:

<Полный ответ>:= <Строка состояния> (<Общий заголовок>|<Заголовок ответа>|<Заголовок ресурса>) <символ новой строки>[<тело ресурса>]

Строка состояния состоит из версии протокола, кода возврата и краткого описания этого кода. Например, она может выглядеть так:

"HTTP/1.0 200 Success".

Заголовок ответа сервера может состоять из адреса URI запрашиваемого ресурса, и/или наименования программы сервера, и/или кода идентификации для работы в защищенном режиме. Состав полей заголовка ресурса является общим и для запроса клиента и для ответа сервера, и состоит из разрешения на метод доступа, типа кодировки тела ресурса (содержания ресурса), длины тела ресурса, типа ресурса, время действия данной копии ресурса, времени последнего изменения ресурса и расширения заголовка.

6.10 Язык гипертекстовой разметки HTML

Язык гипертекстовой разметки HTML (HyperText Markup Language) был предложен Тимом Бернерсом-Ли в 1989 году в качестве одного из компонентов технологии разработки распределенной гипертекстовой системы World Wide Web /28/. Разработчики HTML пытались решить две задачи:

− дать дизайнерам гипертекстовых баз данных простое средство создания документов;

− сделать это средство достаточно мощным, чтобы отразить имевшиеся на тот момент представления об интерфейсе пользователя гипертекстовых баз данных.

Первая задача была решена за счет выбора теговой модели описания документа. Такая модель широко применяется в системах подготовки документов для печати. Примером такой системы является хорошо известный

язык разметки научных документов TeX, предложенный Американским Математическим Обществом, и программы его интерпретации.

К моменту создания HTML существовал стандарт языка разметки печатных документов - SGML (Standard Generalized Markup Language), который и был взят в качестве основы HTML. Предполагалось, что такое решение поможет использовать существующее программное обеспечение для интерпретации нового языка. Однако, будучи доступным широкому кругу пользователей Internet, HTML зажил своей собственной жизнью. Вероятно, многие администраторы баз данных WWW и разработчики программного обеспечения для этой системы имеют довольно смутное представление о стандартном языке разметки SGML.

Вторым важным моментом, повлиявшим на судьбу HTML, стал выбор в качестве элемента гипертекстовой базы данных обычного текстового файла, который хранится средствами файловой системы операционной среды компьютера. Такой выбор был сделан под влиянием следующих факторов:

– такой файл можно создать в любом текстовом редакторе на любой аппаратной платформе в среде любой операционной системы.

– к моменту разработки HTML существовал американский стандарт для разработки сетевых информационных систем - Z39.50, в котором в качестве единицы хранения указывался простой текстовый файл в кодировке LATIN1, что соответствует US ASCII.

Таким образом, гипертекстовая база данных в концепции WWW - это набор текстовых файлов, написанных на языке HTML, который определяет форму представления информации (разметка) и структуру связей этих файлов (гипертекстовые ссылки).

Такой подход предполагает наличие еще одной компоненты технологии - интерпретатора языка. В World Wide Web функции интерпретатора разделены между сервером гипертекстовой базы данных и интерфейсом пользователя.

Сервер, кроме доступа к документам и обработки гипертекстовых ссылок, осуществляет также препроцессорную обработку документов, в то время как интерфейс пользователя осуществляет интерпретацию конструкций языка, связанных с представлением информации.

Принципы построения и интерпретации HTML. *Теговая модель* описывает документ как совокупность элементов, каждый из которых окружен тегами. По своему значению теги близки к понятию скобок "begin/end" в универсальных языках программирования, которые задают области действия имен локальных переменных и т.п. Теги определяют область действия правил интерпретации текстовых элементов документа. Типичным примером такого рода является тег стиля Italic, который определяет область отображения курсива.

Текст на языке HTML:

Текст, следующий за словом "Italic" <I>отображается как курсив</I>.

Текст отображаемый программой интерпретации:

Текст, следующий за словом "Italic" *отображается как курсив*.

В приведенном выше примере элемент текста, который должен быть выделен курсивом, заключен между тегом начала стиля "Italic" - <I> и тегом конца стиля - </I>. Общая схема построения элемента текста в формате HTML может быть записана в следующем виде:

"элемент" := <"имя элемента" "список атрибутов">

содержание элемента </"имя элемента">

Конструкция перед содержанием элемента называется тегом начала элемента, а конструкция, расположенная после содержания элемента, - тегом конца элемента.

Структура гипертекстовой сети задается гипертекстовыми ссылками. Гипертекстовая ссылка - это адрес другого HTML документа, который тематически, логически или каким-либо другим способом связан с документом, в котором ссылка определена.

Для записи гипертекстовых ссылок в системе WWW была разработана форма Universe Resource Locator. Типичным примером использования этой записи можно считать следующий пример:

Этот текст содержит

<A HREF="http://polyn.net.kiae.su/altai/index.html">

гипертекстовую ссылку</A>.

В приведенном выше примере элемент "A", который в HTML называют якорем (anchor), использует атрибут "HREF", который обозначает гипертекстовую ссылку (Hypertext REFerence), для записи этой ссылки в форме URL. Данная ссылка указывает на документ с именем "index.html" в директории "altai" на сервере "polyn.net.kiae.su", доступ к которому осуществляется по протоколу "http".

Гипертекстовые ссылки в HTML делятся на два класса: контекстные гипертекстовые ссылки и общие. Контекстные ссылки вмонтированы в тело документа, как это было продемонстрировано в предыдущем примере, в то время как общие ссылки связаны со всем документом в целом и могут быть использованы при просмотре любого фрагмента документа. Оба класса ссылок присутствуют в стандарте языка с самого его рождения, однако, первоначально наибольшей популярностью пользовались контекстные ссылки. Эта популярность привела к тому, что механизм использования общих ссылок практически полностью "атрофировался". Однако по мере стандартизации интерфейса пользователя и стилей представления информации разработчики языка снова вернулись к общим ссылкам и стремятся приспособить их к задачам управления этим интерфейсом.

Структура HTML-документа позволяет использовать вложенные друг в друга элементы. Собственно, сам документ - это один большой элемент с именем "HTML":

<HTML> Содержание документа </HTML>

Сам элемент HTML или гипертекстовый документ состоит из двух частей: заголовка документа (HEAD) и тела документа (BODY):

<HTML>

<HEAD>

Содержание заголовка
</HEAD>
<BODY>
Содержание тела документа
</BODY>
</HTML>

Приведенная выше форма записи определяет классический HTML-документ. Введение в язык HTML фреймов определило еще один шаблон документа:

```
<HTML>
<!--
Author: HTMLed User
Date:   January 21, 1996
-->
<HEAD>
</HEAD>
<FRAMESET COLS="40%,*">
<NOFRAMES>
<BODY>
Sorry there are not a frame support in your browser.
</BODY>
</NOFRAMES>
<FRAMESET ROWS="120,*,60">
<FRAME SRC=banner.htm NAME=banner>
<FRAME SRC="www.htm"  NAME=content>
<FRAME SRC="bottom.htm" NAME=bottom>
</FRAMESET>
<FRAMESET ROWS="100%">
<FRAME SRC="www_hist.htm" NAME=info>
</FRAMESET>
</FRAMESET>
</HTML>
```

В данном примере представлен документ, который состоит из трех окон внутри рабочего окна программы просмотра, в каждое из которых загружается обычный документ.

Рассмотрим пример классического документа:

```
<HTML>
<!--
Author: Pavel Khramtsov
Date:   January 21, 1996
-->
<HEAD>
<TITLE>This is a Baner</TITLE>
</HEAD>
<BODY BACKGROUND=www_wall.jpg VLINK=0000FF LINK=FF0000>
```

```
<CENTER>
<TABLE>
<TR><TD><IMG SRC="interne0.jpg"></TD>
<TD CENTER>
<H3>Администрирование Internet</H3>
<I>Центр Информационных Технологий, 1996.</I>
</TD></TR>
</TABLE>
</CENTER>
</BODY>
</HTML>
```

Все, что расположено между <HTML> и </HTML> - это документ. Содержание элемента HEAD определяет заголовок документа, который состоит из двух элементов: TITLE и BASE. Вслед за заголовком начинается тело документа, которое содержит в своих первых строках некоторую вводную информацию и содержание документа, оформленное в виде списка /29/.

Каждый документ в системе World Wide Web имеет свое имя, которое указывается в элементе TITLE заголовка документа. Его мы видим в первой строке интерфейса.

Контейнер BODY открывает тело документа. В качестве фона в этом элементе определена картинка back.gif. Эта картинка - "back.gif" - задана частичной формой спецификации URL, которая не задает полного адреса ресурса в сети. Затем мы определили таблицу, состоящую из двух ячеек. В одной ячейке картинка, в то время как в другой - текстовый фрагмент. Текст определен как заголовок третьего уровня, который должен отображаться стилем Italic.

6.11 Контрольные вопросы

1) Система доменных имен.
2) Почта в Интернет.
3) Протокол FTP.
4) Протокол HTTP.
5) Принципы построения и интерпретации HTML.

6.12 Тесты

1) Протокол отправки писем:
   а) FTP;
   б) HTTP;
   в) POP3;
   г) SMTP;
   д) SMTP, POP3.

2) По какому принципу строится система доменных адресов?

а) иерархический;
б) параллельный;
в) последовательный;
г) все правильно;
д) все не правильно.

3) Что входит в основу telnet?
а) концепция сетевого виртуального терминала (Network Virtual Terminal) или NVT;
б) принцип договорных опций (согласование параметров взаимодействия);
в) симметрия связи “терминал-процесс”;
г) все не входят;
д) все входят.

4) В команде ftpd [-d] [-1] [-t timeout] опция d означает:
а) опция отладки;
б) опция автоматической идентификации пользователя;
в) опция d в этой команде отсутствует.

5) Какой из принципов построения адресов www лишний:
а) расширяемость;
б) полнота;
в) читаемость;
г) уникальность.

6) Какой и ниже перечисленных методов позволяет получить данные возвращаемые web-сервером в форме URL?
а) GET;
б) POST;
в) HEAD.

7) В строке ответа сервера "HTTP/1.0 …Succes" вместо многоточия укажите код успешного возврата.
а) 200;
б) 201;
в) 400;
г) 440.

8) Основоположник Интернет:
а) Бил Гейтс;
б) Дел Миллер;
в) Тим Бернерс-Ли.

9) Комментарии в HTML обозначаются:

а) <!--
б) //
в) {
г) /* --*/

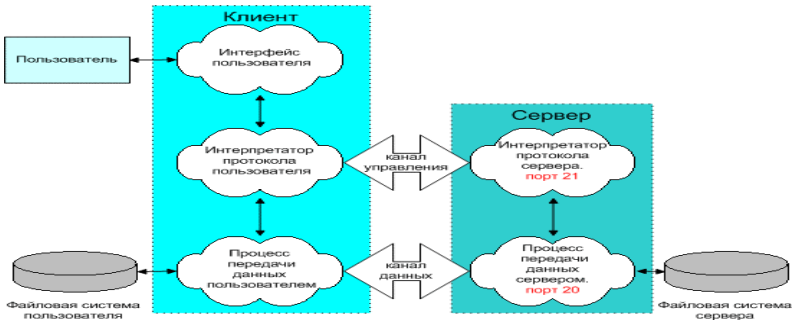
10) Метод для передачи данных на сервер
а) GET
б) POST
в) HEAD

11) Какой день считается днем рождения сети Интернет:
а) 8 марта 1958 года;
б) 23 февраля 1975 года;
в) 1 января 1983 года;
г) 7 ноября 1991 года.

12) Сколько существует схем адресации ресурсов в Internet
а) 8 и более;
б) 2;
в) 6;
г) 3.

13) В команде ftp [-v] [-n] [-i] [-d] [-g] [-s:*имя_файла*] [-a] [-w:*размер*] [*компьютер*] опция v означает:
а) отменяет автоматическое подключение при начальном соединении;
б) отменяет вывод на экран ответа удаленного сервера;
в) отключает подтверждение при передаче нескольких файлов;
г) включает отладочный режим, в котором на экран будут выводиться все команды ftp, передаваемые между клиентом и сервером.

14) Какой протокол изображен на рисунке?



а) HTTP;
б) FTP ;
в) POP3;
г) SMTP.

15) Номер порта протокола TCP/IP для telnet.

а) 20;

б) 30;

в) 23;

г) 80.

Заключение

Данное пособие рассматривает архитектуру взаимосвязанных сетей и обьясняет принципы и протоколы, которые позволяют функционировать таким взаимосвязанным архитектурам как одна единая коммуникационная система. Все пособие посвящено понятию межсетевого обмена, в общем, и технологии Интернета TCP/IP в частности.

Пособие рассматривает как архитектуру сетевых взаимодействий, так и межсетевые коммуникационные средства, и протоколы, требуемые для обеспечения этих средств.

Изучения данного пособия поможет читателю понимать, как возможно взаимное соединение нескольких физических сетей в одну координированную систему, как межсетевые протоколы работают в такой среде, и как прикладные программы используют получившуюся систему.

Список использованных источников

1 http://www.citforum.ru

2 Компьютерные сети: Учебный курс/Пер, с англ.-М.: ТОО «Channel Trading Ltd», 1997. - 696 с.

3 Н.Олиффер, В.Олиффер Компьютерные сети. С-Пб., 1999 г.

4 Норенков И.П., Трудоношин В.А. Телекоммуникационные технологии и сети. - М.: МГТУ им. Н.Э.Баумана, 2000.

5 Кульгин М. Технологии корпоративных сетей: Энциклопедия. - СПб.: Издательство "Питер", 2000. - 704 с.

6 Новиков Ю.В., Кондратенко С.В. Локальные сети: архитектура, алгоритмы, проектирование.- М.: Издательство ЭКОМ, 2000.- 312 с.

7 Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. - СПб.: Издательство "Питер", 2000. - 672 с.

8 Пятибратов А.П. и др. Вычислительные системы, сети и телекоммуникации: Учебник/ Под редакцией А.П. Пятибратова. – М.: Финансы и статистика, 2001. – 512 с.

9 Гук М. Аппаратные средства локальных сетей: Энциклопедия.- СПб.: Издательство "Питер", 2000. - 576 с.

10 Левин М. Как стать системным администратором: Самоучитель. – М.: Познавательная книга плюс, 2001. –320 с.

11 Советов Б.Я., Яковлев С.А. Построение сетей интегрального обслуживания. - Л.: Машиностроение, 1990. - 332 с.

12 Кульгин М. Практика построения компьютерных сетей. Для профессионалов. – СПб.: Питер, 2001. – 320 с.

13 Уолрэнд Дж. Телекоммуникационные и компьютерные сети: Вводный курс / Пер. с англ.- М.: Постмаркет, 2001.- 480с.

14 Нанс Бэрри. Компьютерные сети. — М.: Бином, 1996.

15 Кульгин М. Технологии корпоративных сетей. Энциклопедия. — СПб.: Изд-во «Питер», 1999.

16 Назаров С.В. Администрирование локальных сетей Windows NT: Учеб.пособие. – М.: Финансы и статистика, 2001. – 336 с.

17 http://lemoi-www.dvgu.ru/lect/protoc/tcpip/comer/pref.htm

18 http://lemoi-www.dvgu.ru/lect/protoc/tcpip/tcp/tcp.htm

19 http://lemoi-www.dvgu.ru/lect/protoc/tcpip/networks/contents.htm

20 http://bars.net.ua/

21 Сетевые средства Windows NT. — С.-Пб: BHV-Санкт-Петербург, 1996.

22 Ю.Блэк. Сети ЭВМ: протоколы, стандарты, интерфейсы.- М.: Мир, 1990.

23 Барфилд, Эд , Уолтерс, Брайен. Программирование "клиент-сервер" в локальных вычислительных сетях;Учебник:Пер.с англ. . -М.:Филинъ,1997-423с.

24 Олифер В.Г., Олифер Н.А. Новые технологии и оборудование IP-сетей. - СПб.: БХВ - Санкт-Петербург, 2000. - 512 с.

25 Муштоватый И.Ф. Самоучитель по работе в Интернете/ Под общ. редакцией М.И. Монастырского. – Ростов н/Д.: “Феникс”, 2001. – 320с.

26 Семенов Юрий Алексеевич. Протоколы и ресурсы Internet . -М.: Радио и связь,1996-320с.: ил.

27 http://ipm.kstu.ru/internet/

28 Основы Web-технологий / П.Б. Храмцов, С.А. Брик, А.М. Русак, А.И. Сурин /Под. редакцией П.Б. Храмцова. – М.: ИНТУИТ.РУ ”Интернет-Университет Информационных Технологий”, 2003. – 512 с.

29 Пауэлл Т.А. Полное руководство по HTML / Пер. с англ. А.В. Качанов. – Мн.: ООО “Попурри”, 2001. – 912 с.

Приложение А
(справочное)

Ключ к тестам

Глава 1	Глава 2	Глава 3	Глава 4	Глава 5	Глава 6
1. а	1.г	1.а	1.а	1.г	1.г
2. а	2.в	2.а	2.б	2.а	2.а
3. а	3.г	3.а	3.а	3.г	3.д
4. в	4.в	4.а	4.б	4.б	4.а
5. а,в	5.в,г	5.б	5.в	5.а	5.г
6. б	6.а,в	6.б	6.а	6.в	6.а
7. а	7.в	7.г	7.б	7.б	7.а
8. б	8.б,в	8.г	8.а	8.г	8.в
9. в	9.а	9.г	9.а	9.г	9.а
10.в	10.в	10.а	10.а	10.а	10.б
11.а	11.а,в	11.г	11.а	11.г	11.в
12.б	12.а	12.в	12.а	12.а	12.а
13.б	13.б,г	13.в	13.б	13.г	13.б
14.а	14.а,в	14.а	14.б	14.в	14.б
15.б	15.г	15.г	15.б	15.б	15.в