Министерство образования Республики Беларусь

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

Институт повышения квалификации
и переподготовки кадров

Кафедра «Информатика»

КОМПЬЮТЕРНЫЕ СЕТИ

КУРС ЛЕКЦИЙ
по одноименной дисциплине
для слушателей специальностей 1-40 01 74 «Web-дизайн и компьютерная графика» и 1-40 01 73 «Программное обеспечение информационных систем» заочной формы обучения

Гомель 2015

УДК 004.7(075.8)
ББК 32.971.35я73
К63

Рекомендовано кафедрой «Профессиональная переподготовка»
ИПК и ПК ГГТУ им. П. О. Сухого
(*протокол № 3 от 06.11.2014 г.*)

Составитель А. Н. Осипенко

Рецензент: зав. каф. «Информатика» ГГТУ им. П. О. Сухого канд. физ.-мат. наук *Т. В. Тихоненко*

К63 **Компьютерные** сети : курс лекций по одноим. дисциплине для слушателей специальностей 1-40 01 74 «Web-дизайн и компьютерная графика» и 1-40 01 73 «Программное обеспечение информационных систем» заоч. формы обучения / сост. А. Н. Осипенко. – Гомель : ГГТУ им. П. О. Сухого, 2015. – 169 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц ; 32 Mb RAM ; свободное место на HDD 16 Mb ; Windows 98 и выше ; Adobe Acrobat Reader. – Режим доступа: https://elib.gstu.by. – Загл. с титул. экрана.

Содержит основные теоретические сведения, необходимые для понимания принципов функционирования компьютерных сетей, а также для организации эффективной работы в компьютерных сетях и их наиболее рационального использования.

Для слушателей специальностей 1-40 01 74 «Web-дизайн и компьютерная графика» и 1-40 01 73 «Программное обеспечение информационных систем» заочной формы обучения ИПК и ПК.

УДК 004.7(075.8)
ББК 32.971.35я73

1 Принципы построения компьютерных сетей

1.1 История развития компьютерных сетей

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

В начале 60-х годов двадцатого столетия начали развиваться *интерактивные (с вмешательством пользователя в протекание вычислительного процесса) многотерминальные системы разделения времени.* В таких системах мощный центральный компьютер (*мэйнфрейм*) отдавался в распоряжение нескольким пользователям. Каждый пользователь получал в свое распоряжение *терминал* (монитор с клавиатурой без системного блока), с помощью которого он мог вести диалог с компьютером. Компьютер по очереди обрабатывал программы и данные, поступающие с каждого терминала. Поскольку время реакции компьютера на запрос каждого терминала было достаточно мало, то пользователи практически не замечали параллельную работу нескольких терминалов и у пользователей создавалась иллюзия монопольного пользования компьютером.

Терминалы, как правило, рассредоточивались по всему предприятию и функции ввода-вывода информации были распределенными, но обработка информации проводилась только центральным компьютером. Такие многотерминальные централизованные системы внешне напоминали *локальные вычислительные сети,* до создания которых в действительности нужно было пройти еще большой путь. Сдерживающим фактором для развития компьютерных сетей был в первую очередь экономический фактор. Из-за высокой в то время стоимости компьютеров предприятия не могли позволить себе роскошь купить несколько компьютеров, а значит и объединять в вычислительную сеть было нечего.

Развитие компьютерных сетей началось с решение более простой задачи- доступ к компьютеру с терминалов, удаленных от него на многие сотни, а то и тысячи километров. Терминалы в этом случае соединялись с компьютером через телефонные сети с

помощью специальных устройств *модемов*. Следующим этапом в развитии компьютерных сетей стали соединения через модем не только «*терминал – компьютер*», но и *«компьютер- компьютер»*. Компьютеры получили возможность обмениваться данными в автоматическом режиме, что является базовым механизмом любой компьютерной сети. Тогда впервые появились в сети возможности обмена файлами, синхронизация баз данных, использования электронной почты, т.е. те службы, являющимися в настоящее время традиционными сетевыми сервисами. Такие компьютерные сети получили название *глобальных компьютерных сетей*.

Исторически первые компьютерные сети были созданы агентством ARPA по заданию военного ведомства США. В 1964 году была разработана концепция и архитектура первой в мире компьютерной сети ARPANET, в 1967 впервые было введено понятие *протокола* компьютерной сети. В сентябре 1969 года произошла передача первого компьютерного сообщения между компьютерными узлами Калифорнийского и Стендфордского университетов. В 1977 году сеть ARPANET насчитывала уже 111 узлов, а в 1983 – 4000. Сеть ARPANET прекратила свое существование в 1989 году, не выдержав конкуренцию с набирающей силу сетью Интернет.

В начале 70-х годов двадцатого столетия, благодаря развитию микроэлектроники, были созданы мини-компьютеры, которые стали реальными конкурентами мэйнфреймам. Несколько десятков мини-компьютеров выполняли задачи быстрее одного мэйнфрейма, но при этом все вместе стоили дешевле. Даже небольшие подразделения предприятий получили возможность покупать для себя компьютеры. Мини- компьютеры стали широко использоваться в управлении технологическими процессами, складами, в бухгалтерском учете и т.д. В результате шел интенсивный процесс распределения вычислительных ресурсов по всему предприятию, что, однако, через некоторое время привело к необходимости обратного объединения всех вычислительных ресурсов в одну систему. Теперь это объединение происходило уже не базе одного компьютера, а путем подключения к сети отдельных распределенных компьютеров. Такие компьютерные сети стали называться *локальными компьютерными сетями.*

На начальном этапе создания локальных компьютерных сетей для объединения компьютеров использовались самые разнообразные не стандартизованные устройства и программное обеспечение.

Создание сети в это время требовало от разработчиков больших изобретательности и усилий. В середине 80-х годов положение дел в локальных компьютерных сетях стало кардинально меняться в сторону создания стандартных технологий объединения компьютеров в единую сеть. Были разработаны специальные методы и правила обмена информацией между компьютерами, среди которых наиболее известными стали стандарты Ethernet, TokingRing, FDDI, Arcnet. В указанных стандартах были строго регламентированы длина, вид и порядок следования кодов, посылаемых компьютерами в сеть, правила доступа к сети отдельными компьютерами и т.д. Кроме этого в это время интенсивно начали использоваться стандартные персональные компьютеры, которые очень быстро вытеснили мини-компьютеры и мэйнфреймы. Разработанные стандартные сетевые технологии, а так же использование персональных компьютеров значительно упростили процесс создания компьютерных сетей. Для создания сети достаточно стало приобрести специальные сетевые платы (сетевые адаптеры) соответствующего стандарта, например, Ethernet, стандартный кабель со стандартными разъемами и установить на компьютер одну из популярных сетевых операционных систем, например, NetWare. Присоединение каждого нового компьютера к сети не стало вызывать больших трудностей.

Появление локальных компьютерных сетей внесло много нового в использование вычислительной техники. Появилась возможность быстрого доступа к разделяемым вычислительным ресурсам, к базе данных сразу несколькими пользователями, причем пользователь использовал на своем сетевом компьютере те же знакомые команды, как и при работе с отдельным компьютером. Задачу обработки этих команд и распределения задач между отдельными компьютерами взяла на себя сетевая операционная система.

В настоящее время разделение компьютерных сетей на глобальные и локальные происходит в первую очередь по признаку их территориального размещения, по механизму установления связей между компьютерами и скорости передачи данных.

1.2 Глобальные и локальные сети

Глобальные сети(WAN, WideAreaNetworks) позволяют организовать взаимодействие между компьютерами на больших расстояниях. Эти сети работают на относительно низких скоростях и

могут вносить значительные задержки в передачу информации. Протяженность глобальных сетей может составлять тысячи километров и они интегрированы с сетями масштаба страны.

Локальные сети (LAN, LocalAreaNetworks) обеспечивают наивысшую скорость обмена информацией между компьютерами и типичная локальная сеть занимает пространство в одно или несколько зданий. Протяженность локальных компьютерных сетей составляет всего лишь несколько километров.

Сравнительно недавно появились *городские сети* или *сети мегополисов*(MAN, MetropolitanAreaNetworks). Такие сети предназначены для обслуживания территории крупного города – мегаполиса. В то время как локальные сети наилучшим образом подходят для разделения ресурсов на коротких расстояниях и на больших скоростях (до 100 Мбит), а глобальные сети обеспечивают работу на больших расстояниях и с низкой скоростью (56 и 64 Кбит/с и только на магистралях до 2 Мбит/с), то городские сети занимают промежуточное положение: имеют скорость до 45 Мбит/с и связывают локальные сети в масштабах города с возможностью выхода в глобальные сети.

Механизмы передачи данных в глобальных и локальных сетях существенно отличаются. Глобальные сети ориентированы на соединение, т.е. еще до начала передачи данных между компьютерами сети устанавливается соединение, которое подтверждается обменом компьютеров между собой специальными сигналами (кодами). В локальных сетях, как правило, используются методы, не требующие предварительной установки соединения – данные просто передаются в канал связи без подтверждения готовности их принять. В локальных сетях каждый компьютер имеет сетевой адаптер, который достаточно просто соединяет его с каналом передачи. Глобальные сети содержат активные коммутирующие устройства, мощные маршрутизаторы для распределения сообщений и соответствующие развитые службы по обслуживанию сетевого оборудования. Среди глобальных компьютерных сетей в настоящее время наиболее популярной является сеть Internet, которую более подробно рассмотрим ниже.

1.3 Основные программные и аппаратные компоненты сети

Компьютерная сеть – это сложный комплекс взаимосвязанных и согласованно функционирующих программных и аппаратных компонентов. Изучение сети в целом предполагает знание принципов работы ее отдельных элементов:

- компьютеров;
- коммуникационного оборудования;
- операционных систем;
- сетевых приложений.

Весь комплекс программно-аппаратных средств сети может быть описан многослойной моделью. В основе любой сети лежит аппаратный слой стандартизованных компьютерных платформ. В настоящее время в сетях широко и успешно применяются компьютеры различных классов – от персональных компьютеров до мэйнфреймов и супер-ЭВМ. Набор компьютеров в сети должен соответствовать набору разнообразных задач, решаемых сетью.

Второй слой – это коммуникационное оборудование. Хотя компьютеры и являются центральными элементами обработки данных в сетях, в последнее время не менее важную роль стали играть коммуникационные устройства. Кабельные системы, повторители, мосты, коммутаторы, маршрутизаторы и модульные концентраторы из вспомогательных компонентов сети превратились в основные наряду с компьютерами и системным программным обеспечением как по влиянию на характеристики сети, так и по стоимости. Сегодня коммуникационное устройство может представлять собой сложный специализированный мультипроцессор, который нужно конфигурировать, оптимизировать и администрировать. Изучение принципов работы коммуникационного оборудования требует знакомства с большим количеством протоколов, используемых как в локальных, так и глобальных сетях.

Третьим слоем, образующим программную платформу сети, являются операционные системы (ОС). От того, какие концепции управления локальными и распределенными ресурсами положены в основу сетевой ОС, зависит эффективность работы всей сети. При проектировании сети важно учитывать, насколько просто данная операционная система может взаимодействовать с другими ОС сети, насколько она обеспечивает безопасность и защищенность данных, до какой степени она позволяет наращивать число пользователей, можно

ли перенести ее на компьютер другого типа и многие другие соображения.

Самым верхним слоем сетевых средств являются различные сетевые приложения, такие как сетевые базы данных, почтовые системы, средства архивирования данных, системы автоматизации коллективной работы и др. Очень важно представлять диапазон возможностей, предоставляемых приложениями для различных областей применения, а также знать, насколько они совместимы с другими сетевыми приложениями и операционными системами.

1.4 Что дает предприятию использование сетей

Использование территориально распределенных вычислительных систем соответствует *распределенному характеру прикладных задач* в некоторых предметных областях, таких как автоматизация технологических процессов, банковская деятельность и т. п. Во всех этих случаях имеются рассредоточенные по некоторой территории отдельные потребители информации – сотрудники, организации или технологические установки. Эти потребители достаточно автономно решают свои задачи, поэтому рациональнее предоставлять им собственные вычислительные средства, но в то же время, поскольку решаемые ими задачи тесно взаимосвязаны, их вычислительные средства должны быть объединены в единую систему. Адекватным решением в такой ситуации является использование компьютерной сети.

Для пользователя, кроме выше названных, распределенные системы дают еще и такие преимущества, как *возможность совместного использования данных и устройств*, а также возможность гибкого распределения работ по всей системе. Такое разделение дорогостоящих периферийных устройств – таких как дисковые массивы большой емкости, цветные принтеры, графопостроители, модемы, оптические диски – во многих случаях является основной причиной развертывания сети на предприятии. Пользователь современной вычислительной сети работает за своим компьютером, часто не отдавая себе отчета в том, что при этом он пользуется данными другого мощного компьютера, находящегося за сотни километров от него. Он отправляет электронную почту через модем, подключенный к коммуникационному серверу, общему для нескольких отделов его предприятия. У пользователя создается

иллюзия, что эти ресурсы подключены непосредственно к его компьютеру или же «почти» подключены, так как для их использования нужны незначительные дополнительные действия по сравнению с использованием действительно собственных ресурсов. Такое свойство называется *прозрачностью* сети.

В последнее время стал преобладать другой побудительный мотив развертывания сетей, гораздо более важный в современных условиях, чем экономия средств за счет разделения между сотрудниками корпорации дорогой аппаратуры или программ. Этим мотивом стало стремление обеспечить сотрудникам *оперативный доступ к обширной корпоративной информации*. В условиях жесткой конкурентной борьбы в любом секторе рынка выигрывает, в конечном счете, та фирма, сотрудники которой могут быстро и правильно ответить на любой вопрос клиента – о возможностях их продукции, об условиях ее применения, о решении любых возможных проблем и т. п. В большой корпорации вряд ли даже хороший менеджер может знать все тонкости каждого из выпускаемых фирмой продуктов, тем более что их номенклатура обновляется сейчас каждый квартал, если не месяц. Поэтому очень важно, чтобы менеджер имел возможность со своего компьютера, подключенного к корпоративной сети, скажем в Магадане, передать вопрос клиента на сервер, расположенный в центральном отделении предприятия в Гомеле, и оперативно получить качественный ответ, удовлетворяющий клиента. В этом случае клиент не обратится к другой фирме, а будет пользоваться услугами данного менеджера и впредь.

Чтобы такая работа была возможна, необходимо не только наличие быстрых и надежных связей в корпоративной сети, но и наличие структурированной информации на серверах предприятия, а также возможность эффективного поиска нужных данных. Этот аспект сетевой работы всегда был узким местом в организации доставки информации сотрудникам – даже при существовании мощных СУБД информация в них попадала не самая «свежая» и не в том объеме, который был нужен. В последнее время в этой области наметился некоторый прогресс, связанный с использованием гипертекстовой информационной службы WWW – так называемой технологии *intranet*. Эта технология поддерживает достаточно простой способ представления текстовой и графической информации в виде гипертекстовых страниц, что позволяет быстро поместить

самую свежую информацию на WWW-серверы корпорации. Кроме того, она унифицирует просмотр информации с помощью стандартных программ – Web-броузеров, работа с которыми несложна даже для неспециалиста. Сейчас многие крупные корпорации уже перенесли огромные кипы своих документов на страницы WWW-серверов, и сотрудники этих фирм, разбросанные по всему миру, используют информацию этих серверов через Internet или intranet. Получая легкий и более полный доступ к информации, сотрудники принимают решение быстрее, и качество этого решения, как правило, выше.

Использование сети приводит к *совершенствованию коммуникаций*, то есть к улучшению процесса обмена информацией и взаимодействия между сотрудниками предприятия, а также его клиентами и поставщиками. Сети снижают потребность предприятий в других формах передачи информации, таких как телефон или обычная почта. Зачастую именно возможность организации электронной почты является основной причиной и экономическим обоснованием развертывания на предприятии вычислительной сети. Все большее распространение получают новые технологии, которые позволяют передавать по сетевым каналам связи не только компьютерные данные, но голосовую и видеоинформацию. Корпоративная сеть, которая интегрирует данные и мультимедийную информацию, может использоваться для организации аудио и видеоконференций, кроме того, на ее основе может быть создана собственная внутренняя телефонная сеть.

Конечно, компьютерные сети имеют и свои проблемы. Эти проблемы в основном связаны с организацией эффективного взаимодействия отдельных частей распределенной системы.

Во-первых, это сложности, связанные с программным обеспечением – операционными системами и приложениями. Программирование для распределенных систем принципиально отличается от программирования для централизованных систем. Так, сетевая операционная система, выполняя в общем случае все функции по управлению локальными ресурсами компьютера, сверх того решает многочисленные задачи по предоставлению сетевых служб. Разработка сетевых приложений осложняется из-за необходимости организовать совместную работу их частей, выполняющихся на разных машинах. Много забот доставляет обеспечение совместимости программного обеспечения.

Во-вторых, много проблем связано с транспортировкой сообщений по каналам связи между компьютерами. Основные задачи здесь –обеспечение надежности (чтобы передаваемые данные не терялись и не искажались) и производительности (чтобы обмен данными происходил с приемлемыми задержками). В структуре общих затрат на компьютерную сеть расходы на решение «транспортных вопросов» составляют существенную часть, в то время как в централизованных системах эти проблемы полностью отсутствуют.

В-третьих, это вопросы, связанные с обеспечением безопасности, которые гораздо сложнее решаются в компьютерной сети, чем в централизованной системе. В некоторых случаях, когда безопасность особенно важна, от использования сети лучше вообще отказаться.

Главным доказательством эффективности сетей является бесспорный факт их повсеместного распространения. Трудно найти сколь-нибудь крупное предприятие, на котором не было хотя бы односегментной сети персональных компьютеров; все больше и больше появляется крупных сетей с сотнями рабочих станций и десятками серверов, некоторые большие организации и предприятия обзаводятся частными глобальными сетями, объединяющими их филиалы, удаленные на тысячи километров.

1.5Основные проблемы построения сетей

При создании компьютерных сетей их разработчикам пришлось решить много проблем. Далее мы рассмотрим только наиболее важные из них, причем в той последовательности, в которой они естественно возникали в процессе развития и совершенствования сетевых технологий.

Механизмы взаимодействия компьютеров в сети многое позаимствовали у схемы взаимодействия компьютера с периферийными устройствами, поэтому начнем рассмотрение принципов работы сети с этого «досетевого» случая.

1.5.1 Связь компьютера с периферийными устройствами.

Для обмена данными между компьютером и периферийным устройством (ПУ) в компьютере предусмотрен внешний *интерфейс* (рисунок 1.1), то есть набор проводов, соединяющих компьютер и периферийное устройство, а также набор правил обмена

информацией по этим проводам (иногда вместо термина *интерфейс* употребляется термин *протокол* – подробней об этих важных терминах мы еще поговорим). Примерами интерфейсов, используемых в компьютерах, являются параллельный интерфейс Centronics, предназначенный, как правило, для подключения принтеров, и последовательный интерфейс RS-232C, через который подключаются мышь, модем и много других устройств. В последнее время для этих целей используется универсальный интерфейс USB. Интерфейс реализуется со стороны компьютера совокупностью аппаратных и программных средств: контроллером ПУ и специальной программой, управляющей этим контроллером, которую часто называют *драйвером* соответствующего периферийного устройства.

Со стороны ПУ интерфейс чаще всего реализуется аппаратным устройством управления, хотя встречаются и программно-управляемые периферийные устройства.

Программа, выполняемая процессором, может обмениваться данными с помощью команд ввода/вывода с любыми модулями, подключенными к внутренней шине компьютера, в том числе и с контроллерами ПУ.

Периферийные устройства могут принимать от компьютера как данные, например байты информации, которую нужно распечатать на бумаге, так и команды управления, в ответ на которые ПУ может выполнить специальные действия, например, перевести головку диска на требуемую дорожку или же вытолкнуть лист бумаги из принтера. Периферийное устройство использует внешний интерфейс компьютера не только для приема информации, но и для передачи информации в компьютер, то есть обмен данными по внешнему интерфейсу, как правило, является двунаправленным. Так, например, даже принтер, который по своей природе является устройством вывода информации, возвращает в компьютер данные о своем состоянии.

Контроллеры ПУ принимают команды и данные от процессора в свой внутренний буфер, который часто называется регистром или портом, затем выполняют необходимые преобразования этих данных и команд в соответствии с форматами, понятными ПУ, и выдают их на внешний интерфейс.

Распределение обязанностей между контроллером и драйвером ПУ может быть разным, но обычно контроллер выполняет набор простых команд по управлению ПУ, а драйвер использует эти

команды, чтобы заставить устройство совершать более сложные действия по некоторому алгоритму. Например, контроллер принтера может поддерживать такие элементарные команды, как «Печать символа», «Перевод строки», «Возврат каретки» и т. п. Драйвер же принтера с помощью этих команд организует печать строк символов, разделение документа на страницы и другие более высокоуровневые операции. Для одного и того же контроллера можно разработать различные драйверы, которые будут управлять данным ПУ по-разному – одни лучше, а другие хуже – в зависимости от опыта и способностей программистов, их разработавших.

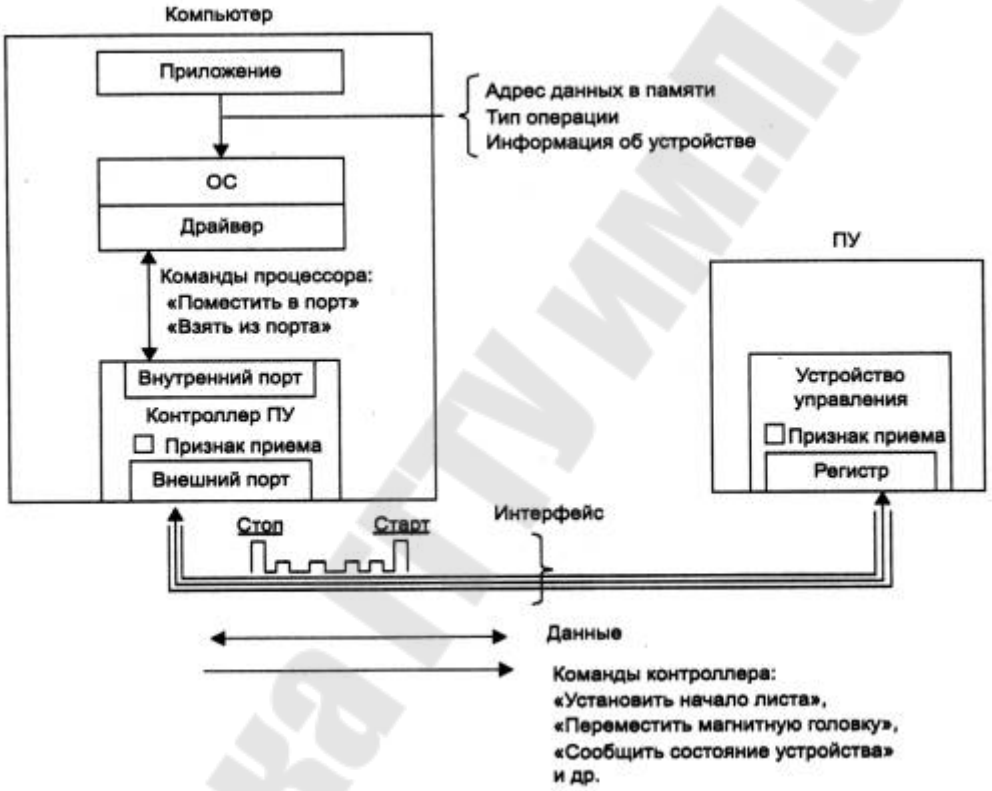


Рисунок 1.1– Связь компьютера с периферийным устройством

Рассмотрим схему передачи одного байта информации от прикладной программы на периферийное устройство. Программа, которой потребовалось выполнить обмен данными с ПУ, обращается к драйверу этого устройства, сообщая ему в качестве параметра адрес байта памяти, который нужно передать. Драйвер загружает значение этого байта в буфер контроллера ПУ, который начинает последовательно передавать биты в линию связи, представляя каждый бит соответствующим электрическим сигналом. Чтобы устройству управления ПУ стало понятно, что начинается передача байта, перед передачей первого бита информации контроллер ПУ формирует

стартовый сигнал специфической формы, а после передачи последнего информационного бита – стоповый сигнал. Эти сигналы *синхронизируют* передачу байта.

Кроме информационных бит, контроллер может передавать бит контроля четности для повышения достоверности обмена. Устройство управления, обнаружив на соответствующей линии стартовый бит, выполняет подготовительные действия и начинает принимать информационные биты, формируя из них байт в своем приемном буфере. Если передача сопровождается битом четности, то выполняется проверка правильности передачи: при правильно выполненной передаче в соответствующем регистре устройства управления устанавливается признак завершения приема информации.

Обычно на драйвер возлагаются наиболее сложные функции протокола (например, подсчет контрольной суммы последовательности передаваемых байтов, анализ состояния периферийного устройства, проверка правильности выполнения команды). Но даже самый примитивный драйвер контроллера должен поддерживать как минимум две операции: «Взять данные из контроллера в оперативную память» и «Передать данные из оперативной памяти в контроллер».

Существуют как весьма специализированные интерфейсы, пригодные для подключения узкого класса устройств (например, графических мониторов высокого разрешения), так и интерфейсы общего назначения, являющиеся стандартными и позволяющие подключать различные периферийные устройства. Примером такого интерфейса является интерфейсы RS-232C и USB.

1.5.2 Простейший случай взаимодействия двух компьютеров

В самом простом случае взаимодействие компьютеров может быть реализовано с помощью тех же самых средств, которые используются для взаимодействия компьютера с периферией, например, через последовательный интерфейс RS-232C. В отличие от взаимодействия компьютера с периферийным устройством, когда программа работает, как правило, только с одной стороны – со стороны компьютера, в этом случае происходит взаимодействие двух программ, работающих на каждом из компьютеров.

Программа, работающая на одном компьютере, не может получить непосредственный доступ к ресурсам другого компьютера – его дискам, файлам, принтеру. Она может только «попросить» об

этом программу, работающую на том компьютере, которому принадлежат эти ресурсы. Эти «просьбы» выражаются в виде *сообщений*, передаваемых по каналам связи между компьютерами. Сообщения могут содержать не только команды на выполнение некоторых действий, но и собственно информационные данные (например, содержимое некоторого файла).

Рассмотрим случай, когда пользователю, работающему с текстовым редактором на персональном компьютере А, нужно прочитать часть некоторого файла, расположенного на диске персонального компьютера В (рисунок 1.2). Предположим, что мы связали эти компьютеры по кабелю связи через COM-порты, которые, как известно, реализуют интерфейс RS-232C (такое соединение часто называют нуль-модемным). Пусть для определенности компьютеры работают под управлением MS-DOS, хотя принципиального значения в данном случае это не имеет.

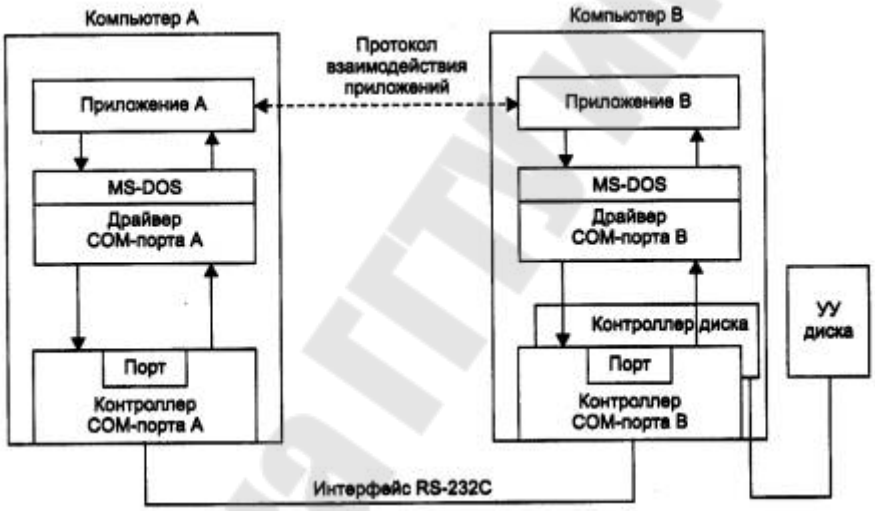


Рисунок 1.2– Взаимодействие двух компьютеров

Драйвер COM-порта вместе с контроллером COM-порта работают примерно так же, как и в описанном выше случае взаимодействия ПУ с компьютером. Однако при этом роль устройства управления ПУ выполняет контроллер и драйвер COM-порта другого компьютера. Вместе они обеспечивают передачу по кабелю между компьютерами одного байта информации. (В «настоящих» локальных сетях подобные функции передачи данных в линию связи выполняются сетевыми адаптерами и их драйверами.)

Драйвер компьютераВ периодически опрашивает признак завершения приема, устанавливаемый контроллером при правильно выполненной передаче данных, и при его появлении считывает

принятый байт из буфера контроллера в оперативную память, делая его тем самым доступным для программ компьютера В. В некоторых случаях драйвер вызывается асинхронно, по прерываниям от контроллера.

Таким образом, в распоряжении программ компьютеров А и В имеется средство для передачи одного байта информации. Но рассматриваемая в нашем примере задача значительно сложнее, так как нужно передать не один байт, а определенную часть заданного файла. Все связанные с этим дополнительные проблемы должны решить программы более высокого уровня, чем драйверы COM-портов. Для определенности назовем такие программы компьютеров А и В приложением А и приложением В соответственно. Итак, приложение А должно сформировать сообщение-запрос для приложения В. В запросе необходимо указать имя файла, тип операции (в данном случае – чтение), смещение и размер области файла, содержащей нужные данные.

Чтобы передать это сообщение компьютеру В, приложение А обращается к драйверу COM-порта, сообщая ему адрес в оперативной памяти, по которому драйвер находит сообщение и затем передает его байт за байтом приложению В. Приложение В, приняв запрос, выполняет его, то есть считывает требуемую область файла с диска с помощью средств локальной ОС в буферную область своей оперативной памяти, а далее с помощью драйвера COM-порта передает считанные данные по каналу связи в компьютер А, где они и попадают к приложению А.

Описанные функции приложения А могла бы выполнить сама программа текстового редактора, но включать эти функции в состав каждого приложения – текстовых редакторов, графических редакторов, систем управления базами данных и других приложений, которым нужен доступ к файлам, – не очень рационально (хотя существует большое количество программ, которые действительно самостоятельно решают все задачи по межмашинному обмену данными, например Kermit – программа обмена файлами через COM-порты, реализованная для различных ОС, NortonCommander 3.0 с его функцией Link). Гораздо выгоднее создать специальный программный модуль, который будет выполнять функции формирования сообщений-запросов и приема результатов для всех приложений компьютера. Как уже было ранее сказано, такой служебный модуль называется клиентом. На стороне же

компьютераВ должен работать другой модуль – сервер, постоянно ожидающий прихода запросов на удаленный доступ к файлам, расположенным на диске этого компьютера. Сервер, приняв запрос из сети, обращается к локальному файлу и выполняет с ним заданные действия, возможно, с участием локальной ОС.

Программные клиент и сервер выполняют системные функции по обслуживанию запросов приложений компьютера А на удаленный доступ к файлам компьютера В. Чтобы приложения компьютера В могли пользоваться файлами компьютера А, описанную схему нужно симметрично дополнить клиентом для компьютера В и сервером для компьютера А.

Схема взаимодействия клиента и сервера с приложениями и операционной системой приведена на рисунке 1.3. Несмотря на то, что мы рассмотрели очень простую схему аппаратной связи компьютеров, функции программ, обеспечивающих доступ к удаленным файлам, очень похожи на функции модулей сетевой операционной системы, работающей в сети с более сложными аппаратными связями компьютеров.

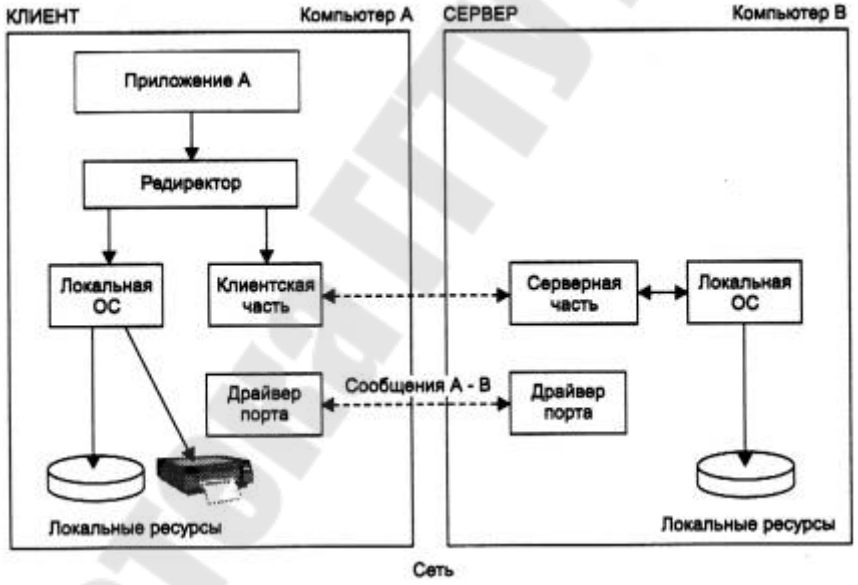


Рисунок 1.3– Взаимодействие программных компонентов при связи двух компьютеров

Очень удобной и полезной функцией клиентской программы является способность отличить запрос к удаленному файлу от запроса к локальному файлу. Если клиентская программа умеет это делать, то приложения не должны заботиться о том, с каким файлом они работают (локальным или удаленным), клиентская программа сама

распознает и *перенаправляет (redirect)* запрос к удаленной машине. Отсюда и название, часто используемое для клиентской части сетевой ОС, *–редиректор*. Иногда функции распознавания выделяются в отдельный программный модуль, в этом случае редиректором называют не всю клиентскую часть, а только этот модуль.

1.5.3 Проблемы стандартизации компьютерных сетей. Понятия интерфейса, протокола и стека

По своей сущности компьютерная сеть является совокупностью компьютеров и сетевого оборудования, соединенных каналами связи. Поскольку компьютеры и сетевое оборудование могут быть разных производителей, то возникает проблема их совместимости. Без принятия всеми производителя общепринятых правил построения оборудования создание компьютерной сети было бы невозможно. Поэтому разработка и создание компьютерных сетей может происходить только в рамках утвержденных *стандартов.*

В основу стандартизации компьютерных сетей положен принцип декомпозиции, т.е. разделения сложных задач на отдельные более простые подзадачи. Каждая подзадача имеет четко определенные функции и строго установленные связи между подзадачами. При более внимательном рассмотрении работы компьютера в сети можно выделить две основные подзадачи:

- взаимодействие программного обеспечения пользователя с физическим каналом связи (посредством сетевой карты) в пределах одного компьютера;
- взаимодействие компьютера через канал связи с другим компьютером.

Современное программное обеспечение компьютера имеет многоуровневую модульную структуру, т.е. программный код, написанный программистом и видимый на экране монитора (модуль верхнего уровня), проходит несколько уровней обработки, прежде чем превратится в электрический сигнал (модуль нижнего уровня), передаваемый в канал связи.

При взаимодействии компьютеров через канал связи оба компьютера должны выполнять ряд соглашений. Например, они должны согласовать величину и форму электрических сигналов, длину сообщений, методы контроля достоверности и т.д. Соглашения должны быть такими, чтобы они были поняты каждым модулем на соответствующе уровне каждого компьютера.

Суть работы многоуровневого протокола можно пояснить как «письмо в конверте». Каждый уровень протокола надписывает на «конверте» свою информацию. Сетям нужно только понимать «надпись» на «конверте», чтобы предать его в место назначения, а до содержания письма им дела нет.

На Рисунке1.4 схематически показана модель взаимодействия двух компьютеров в сети. Для упрощения показаны четыре уровня модулей для каждого компьютера. Процедура взаимодействие каждого уровня этих компьютеров может быть описана в виде набора правил взаимодействия каждой пары модулей соответствующих уровней.

Формализованные правила, определяющие последовательность и формат сообщений, которыми обмениваются модули, лежащие на одном уровне (по горизонтали), но в различных компьютерах называются *протоколами*.

Модули, реализующие протоколы двух соседних уровней (по вертикали) и находящиеся в одном компьютере, также взаимодействуют друг с другом в соответствии с четко определенными правилами и с помощью стандартизованных форматов сообщений. Эти правила называются *интерфейсом* и определяют набор сервисов, предоставляемых данным уровнем соседнему уровню.

Другими словами, в сетевых технологиях традиционно принято, что протоколы определяют правила взаимодействия модулей одного уровня, но в разных компьютерах, а интерфейсы – соседних уровней в одном компьютере. Модули, таким образом, должны обрабатывать: во-первых свой собственный протокол, а во-вторых интерфейсы с соседними уровнями.

Иерархически организованный набор протоколов для взаимодействия компьютеров в сети называется *стеком коммуникационных протоколов*.

Коммуникационные протоколы могут быть реализованы как программно, так и аппаратно. Протоколы нижних уровней, как правило, реализуются комбинацией программно-аппаратных средств, а протоколы верхних уровней чисто программными средствами.

Рисунок 1.4– Взаимодействие двух компьютеров в сети

Отметим, что протоколы каждого уровня обладают независимостью друг от друга, т.е. протокол любого уровня может быть изменен, не оказывая при этом никакого влияния на протокол другого уровня. Главное, чтобы интерфейсы между уровнями обеспечивали необходимые связи между ними. В сущности, протокол и интерфейс выражают одно и то же понятие, но традиционно в сетях за ними закрепили разные области действия: протоколы определяют правила взаимодействия модулей одного уровня в разных узлах, а интерфейсы – модулей соседних уровней в одном узле.

Средства каждого уровня должны отрабатывать, во-первых, свой собственный протокол, а во-вторых, интерфейсы с соседними уровнями.

Иерархически организованный набор протоколов, достаточный для организации взаимодействия узлов в сети, называется *стеком коммуникационных протоколов*.

Коммуникационные протоколы могут быть реализованы как программно, так и аппаратно. Протоколы нижних уровней часто реализуются комбинацией программных и аппаратных средств, а

протоколы верхних уровней – как правило, чисто программными средствами.

Программный модуль, реализующий некоторый протокол, часто для краткости также называют «протоколом». При этом соотношение между протоколом – формально определенной процедурой и протоколом – программным модулем, реализующим эту процедуру, аналогично соотношению между алгоритмом решения некоторой задачи и программой, решающей эту задачу.

Понятно, что один и тот же алгоритм может быть запрограммирован с разной степенью эффективности. Точно так же и протокол может иметь несколько программных реализации. Именно поэтому при сравнении протоколов следует учитывать не только логику их работы, но и качество программных решений. Более того, на эффективность взаимодействия устройств в сети влияет качество всей совокупности протоколов, составляющих стек, в частности, насколько рационально распределены функции между протоколами разных уровней и насколько хорошо определены интерфейсы между ними.

Протоколы реализуются не только компьютерами, но и другими сетевыми устройствами – концентраторами, мостами, коммутаторами, маршрутизаторами и т. д. Действительно, в общем случае связь компьютеров в сети осуществляется не напрямую, а через различные коммуникационные устройства. В зависимости от типа устройства в нем должны быть встроенные средства, реализующие тот или иной набор протоколов.

Принцип взаимодействия компьютеров в сети можно объяснить на примере сотрудничества двух фирм (Рис 1.5). Два генеральных менеджера каждой из фирм осуществляют сделки между собой на основании заключенных договоров и соглашений. Указанные взаимодействия являются «протоколом уровня генеральных менеджеров». На каждой из фирм у менеджеров есть секретари, причем каждый менеджер имеет свой метод и стиль работы с секретарем. Один, например, предпочитает устные указания, а второй дает только письменные распоряжения. Таким образом, каждая фирма имеет свой собственный интерфейс «главный менеджер – секретарь», что не мешает, однако, нормально работать генеральным менеджерам между собой. Секретари в свою очередь договорились обмениваться информацией с помощью факсов, реализуя протокол «секретарь – секретарь». В случае, если секретари перейдут на

электронную почту, то генеральные менеджеры этого даже и не заметят – главное, чтобы секретари выполняли их распоряжения, т.е. должен безукоризненно работать интерфейс «менеджер – секретарь». С другой стороны, менеджеры могут заключить совершенно новый договор, т.е. изменить «протокол уровня генеральных менеджеров». Передача не старого, а нового договора на уровне секретарей пройдет для этих секретарей абсолютно незамеченной.

Рисунок 1.5– Пример многоуровневого взаимодействия предприятий

В рассмотренном примере мы определили два уровня протоколов – уровень генеральных менеджеров и уровень секретарей. Каждый из указанных уровней имеет свой собственный протокол, который может быть изменен независимо от протокола другого уровня. Такую независимость обеспечивает правильное функционирование интерфейсов «менеджер – секретарь».

Независимость протоколов каждого уровня друг от друга и взаимодействие самих уровней посредством интерфейсов является важнейшей предпосылкой для создания ряда стандартных протоколов для компьютерных сетей.

1.5.4 Проблемы физической передачи данных

Даже при рассмотрении простейшей сети, состоящей всего из двух машин, можно увидеть многие проблемы, присущие любой компьютерной сети, в том числе проблемы, связанные с физической передачей сигналов по линиям связи, без решения которой невозможен любой вид связи.

В вычислительной технике для представления данных используется двоичный код. Внутри компьютера единицам и нулям данных соответствуют дискретные электрические сигналы. Представление данных в виде электрических или оптических сигналов называется *кодированием*. Существуют различные способы кодирования двоичных цифр 1 и 0, например, потенциальный способ, при котором единице соответствует один уровень напряжения, а нулю – другой, или импульсный способ, когда для представления цифр используются импульсы различной или одной полярности.

Аналогичные подходы могут быть использованы для кодирования данных и при передаче их между двумя компьютерами по линиям связи. Однако эти линии связи отличаются по своим электрическим характеристикам от тех, которые существуют внутри компьютера. Главное отличие внешних линий связи от внутренних состоит в их гораздо большей протяженности, а также в том, что они проходят вне экранированного корпуса по пространствам, зачастую подверженным воздействию сильных электромагнитных помех. Все это приводит к значительно большим искажениям прямоугольных импульсов (например, «заваливанию» фронтов), чем внутри компьютера. Поэтому для надежного распознавания импульсов на приемном конце линии связи при передаче данных внутри и вне компьютера не всегда можно использовать одни и те же скорости и способы кодирования. Например, медленное нарастание фронта импульса из-за высокой емкостной нагрузки линии требует передачи импульсов с меньшей скоростью (чтобы передний и задний фронты соседних импульсов не перекрывались и импульс успел дорасти до требуемого уровня).

В вычислительных сетях применяют как потенциальное, так и импульсное кодирование дискретных данных, а также специфический способ представления данных, который никогда не используется внутри компьютера, – *модуляцию* (рисунок 1.6). При модуляции дискретная информация представляется синусоидальным сигналом той частоты, которую хорошо передает имеющаяся линия связи.

Потенциальное или импульсное кодирование применяется на каналах высокого качества, а модуляция на основе синусоидальных сигналов предпочтительнее в том случае, когда канал вносит сильные искажения в передаваемые сигналы. Обычно модуляция используется в глобальных сетях при передаче данных через аналоговые телефонные каналы связи, которые были разработаны для передачи

голоса в аналоговой форме и поэтому плохо подходят для непосредственной передачи импульсов.

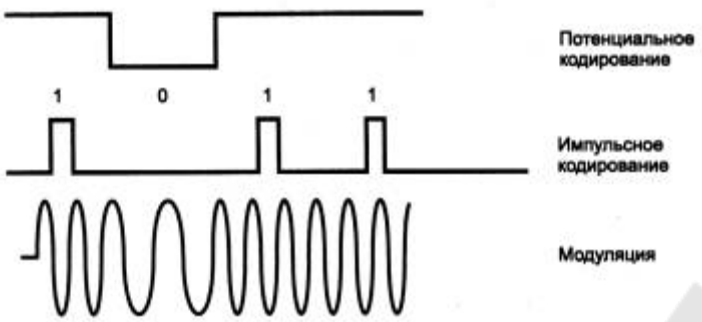


Рисунок 1.6– Примеры представления дискретной информации

На способ передачи сигналов влияет и количество проводов в линиях связи между компьютерами. Для сокращения стоимости линий связи в сетях обычно стремятся к сокращению количества проводов и из-за этого используют не параллельную передачу всех бит одного байта или даже нескольких байт, как это делается внутри компьютера, а последовательную, побитную передачу, требующую всего одной пары проводов.

Еще одной проблемой, которую нужно решать при передаче сигналов, является проблема взаимной *синхронизации* передатчика одного компьютера с приемником другого. При организации взаимодействия модулей внутри компьютера эта проблема решается очень просто, так как в этом случае все модули синхронизируются от общего тактового генератора. Проблема синхронизации при связи компьютеров может решаться разными способами, как с помощью обмена специальными тактовыми синхроимпульсами по отдельной линии, так и с помощью периодической синхронизации заранее обусловленными кодами или импульсами характерной формы, отличающейся от формы импульсов данных.

Несмотря на предпринимаемые меры – выбор соответствующей скорости обмена данными, линий связи с определенными характеристиками, способа синхронизации приемника и передатчика, – существует вероятность искажения некоторых бит передаваемых данных. Для повышения надежности передачи данных между компьютерами часто используется стандартный прием – подсчет *контрольной суммы* и передача ее по линиям связи после каждого байта или после некоторого блока байтов. Часто в протокол обмена

данными включается как обязательный элемент сигнал-квитанция, который подтверждает правильность приема данных и посылается от получателя отправителю.

Задачи надежного обмена двоичными сигналами, представленными соответствующими электромагнитными сигналами, в вычислительных сетях решает определенный класс оборудования. В локальных сетях это *сетевые адаптеры*, а в глобальных сетях – аппаратура передачи данных, к которой относятся, например, устройства, выполняющие модуляцию и демодуляцию дискретных сигналов, – *модемы*. Это оборудование кодирует и декодирует каждый информационный бит, синхронизирует передачу электромагнитных сигналов по линиям связи, проверяет правильность передачи по контрольной сумме и может выполнять некоторые другие операции. Сетевые адаптеры рассчитаны, как правило, на работу с определенной *передающей средой* – коаксиальным кабелем, витой парой, оптоволокном и т. п. Каждый тип передающей среды обладает определенными электрическими характеристиками, влияющими на способ использования данной среды, и определяет скорость передачи сигналов, способ их кодирования и некоторые другие параметры.

1.5.5 Проблемы объединения нескольких компьютеров

До сих пор мы рассматривали вырожденную сеть, состоящую всего из двух машин. При объединении в сеть большего числа компьютеров возникает целый комплекс новых проблем.

Топология физических связей

В первую очередь необходимо выбрать способ организации физических связей, то есть *топологию*. Под топологией вычислительной сети понимается конфигурация графа, вершинам которого соответствуют компьютеры сети (иногда и другое оборудование, например концентраторы), а ребрам – физические связи между ними. Компьютеры, подключенные к сети, часто называют *станциями* или *узлами* сети.

Заметим, что конфигурация *физических связей* определяется электрическими соединениями компьютеров между собой и может отличаться от конфигурации *логических связей* между узлами сети. Логические связи представляют собой маршруты передачи данных между узлами сети и образуются путем соответствующей настройки коммуникационного оборудования.

Выбор топологии электрических связей существенно влияет на многие характеристики сети. Например, наличие резервных связей повышает надежность сети и делает возможным балансирование загрузки отдельных каналов. Простота присоединения новых узлов, свойственная некоторым топологиям, делает сеть легко расширяемой. Экономические соображения часто приводят к выбору топологий, для которых характерна минимальная суммарная длина линий связи. Рассмотрим некоторые, наиболее часто встречающиеся топологии.

Полносвязная топология (рисунок 1.7, а) соответствует сети, в которой каждый компьютер сети связан со всеми остальными. Несмотря на логическую простоту, этот вариант оказывается громоздким и неэффективным. Действительно, каждый компьютер в сети должен иметь большое количество коммуникационных портов, достаточное для связи с каждым из остальных компьютеров сети. Для каждой пары компьютеров должна быть выделена отдельная электрическая линия связи. Полносвязные топологии применяются редко, так как не удовлетворяют ни одному из приведенных выше требований. Чаще этот вид топологии используется в многомашинных комплексах или глобальных сетях при небольшом количестве компьютеров.

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

Ячеистая топология (*mesh*) получается из полносвязной путем удаления некоторых возможных связей (рисунок 1.7, б). В сети с ячеистой топологией непосредственно связываются только те компьютеры, между которыми происходит интенсивный обмен данными, а для обмена данными между компьютерами, не соединенными прямыми связями, используются транзитные передачи через промежуточные узлы. Ячеистая топология допускает соединение большого количества компьютеров и характерна, как правило, для глобальных сетей.

Общая шина (рисунок 1.7, в) является очень распространенной (а до недавнего времени самой распространенной) топологией для локальных сетей. В этом случае компьютеры подключаются к одному коаксиальному кабелю по схеме «монтажного ИЛИ». Передаваемая информация может распространяться в обе стороны. Применение общей шины снижает стоимость проводки, унифицирует

подключение различных модулей, обеспечивает возможность почти мгновенного широковещательного обращения ко всем станциям сети. Таким образом, основными преимуществами такой схемы являются дешевизна и простота разводки кабеля по помещениям. Самый серьезный недостаток общей шины заключается в ее низкой надежности: любой дефект кабеля или какого-нибудь из многочисленных разъемов полностью парализует всю сеть. К сожалению, дефект коаксиального разъема редкостью не является. Другим недостатком общей шины является ее невысокая производительность, так как при таком способе подключения в каждый момент времени только один компьютер может передавать данные в сеть. Поэтому пропускная способность канала связи всегда делится здесь между всеми узлами сети.

Топология *звезда* (рисунок 1.7, г). В этом случае каждый компьютер подключается отдельным кабелем к общему устройству, называемому *концентратором*, который находится в центре сети. В функции концентратора входит направление передаваемой компьютером информации одному или всем остальным компьютерам сети. Главное преимущество этой топологии перед общей шиной – существенно большая надежность. Любые неприятности с кабелем касаются лишь того компьютера, к которому этот кабель присоединен, и только неисправность концентратора может вывести из строя всю сеть. Кроме того, концентратор может играть роль интеллектуального фильтра информации, поступающей от узлов в сеть, и при необходимости блокировать запрещенные администратором передачи.

К недостаткам топологии типа звезда относится более высокая стоимость сетевого оборудования из-за необходимости приобретения концентратора. Кроме того, возможности по наращиванию количества узлов в сети ограничиваются количеством портов концентратора. Иногда имеет смысл строить сеть с использованием нескольких концентраторов, иерархически соединенных между собой связями типа звезда (рисунок 1.7,д). В настоящее время иерархическая звезда является самым распространенным типом топологии связей как в локальных, так и глобальных сетях.

В сетях с *кольцевой* конфигурацией (рисунок 1.7, е) данные передаются по кольцу от одного компьютера к другому, как правило, в одном направлении. Если компьютер распознает данные как «свои», то он копирует их себе во внутренний буфер. В сети с кольцевой

топологией необходимо принимать специальные меры, чтобы в случае выхода из строя или отключения какой-либо станции не прервался канал связи между остальными станциями. Кольцо представляет собой очень удобную конфигурацию для организации обратной связи – данные, сделав полный оборот, возвращаются к узлу-источнику. Поэтому этот узел может контролировать процесс доставки данных адресату. Часто это свойство кольца используется для тестирования связности сети и поиска узла, работающего некорректно. Для этого в сеть посылаются специальные тестовые сообщения.

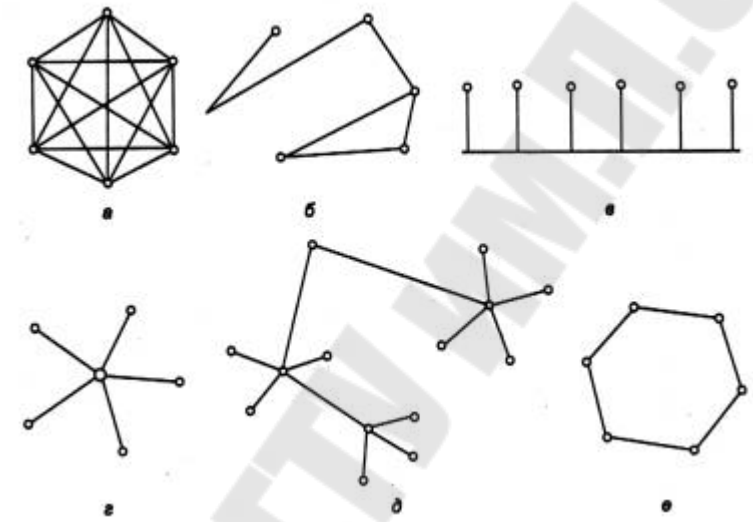


Рисунок 1.7– Типовые топологии сетей

В то время как небольшие сети, как правило, имеют типовую топологию – звезда, кольцо или общая шина, для крупных сетей характерно наличие произвольных связей между компьютерами. В таких сетях можно выделить отдельные произвольно связанные фрагменты (подсети), имеющие типовую топологию, поэтому их называют сетями со *смешанной топологией* (рисунок 1.8).

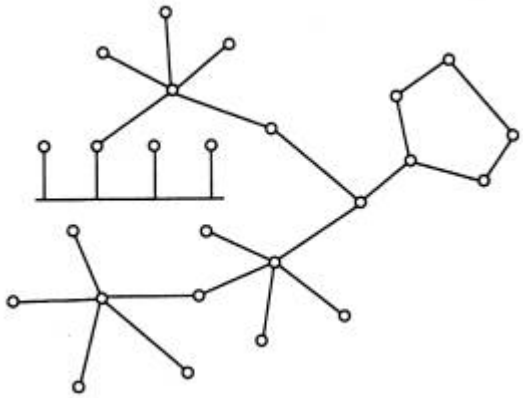


Рисунок 1.8– Смешанная топология

Организация совместного использования линий связи

Только в сети с полносвязной топологией для соединения каждой пары компьютеров имеется отдельная линия связи. Во всех остальных случаях неизбежно возникает вопрос о том, как организовать совместное использование линий связи несколькими компьютерами сети. Как и всегда при разделении ресурсов, главной целью здесь является удешевление сети.

В вычислительных сетях используют как индивидуальные линии связи между компьютерами, так и *разделяемые (shared)*, когда одна линия связи попеременно используется несколькими компьютерами. В случае применения разделяемых линий связи (часто используется также термин разделяемая среда передачи данных – sharedmedia) возникает комплекс проблем, связанных с их совместным использованием, который включает как чисто электрические проблемы обеспечения нужного качества сигналов при подключении к одному и тому же проводу нескольких приемников и передатчиков, так и логические проблемы разделения во времени доступа к этим линиям.

Классическим примером сети с разделяемыми линиями связи являются сети с топологией «общая шина», в которых один кабель совместно используется всеми компьютерами сети. Ни один из компьютеров сети в принципе не может индивидуально, независимо от всех других компьютеров сети, использовать кабель, так как при одновременной передаче данных сразу несколькими узлами сигналы смешиваются и искажаются. В топологиях «кольцо» или «звезда» индивидуальное использование линий связи, соединяющих компьютеры, принципиально возможно, но эти кабели часто также рассматривают как разделяемые для всех компьютеров сети, так что, например, только один компьютер кольца имеет право в данный момент времени отправлять по кольцу пакеты другим компьютерам.

Существуют различные способы решения задачи организации совместного доступа к разделяемым линиям связи. Внутри компьютера проблемы разделения линий связи между различными модулями также существуют – примером является доступ к системной шине, которым управляет либо процессор, либо специальный арбитр шины. В сетях организация совместного доступа к линиям связи имеет свою специфику из-за существенно большего времени распространения сигналов по длинным проводам, к тому же

это время для различных пар компьютеров может быть различным. Из-за этого процедуры согласования доступа к линии связи могут занимать слишком большой промежуток времени и приводить к значительным потерям производительности сети.

Несмотря на все эти сложности, в локальных сетях разделяемые линии связи используются очень часто. Этот подход, в частности, реализован в широко распространенных классических технологиях Ethernet и TokenRing. Однако в последние годы наметилась тенденция отказа от разделяемых сред передачи данных и в локальных сетях. Это связано с тем, что за достигаемое таким образом удешевление сети приходится расплачиваться производительностью.

Сеть с разделяемой средой при большом количестве узлов будет работать всегда медленнее, чем аналогичная сеть с индивидуальными линиями связи, так как пропускная способность индивидуальной линии связи достается одному компьютеру, а при ее совместном использовании – делится на все компьютеры сети.

Часто с такой потерей производительности мирятся ради увеличения экономической эффективности сети. Не только в классических, но и в совсем новых технологиях, разработанных для локальных сетей, сохраняется режим разделяемых линий связи. Например, разработчики технологии GigabitEthernet, принятой в 1998 году в качестве нового стандарта, включили режим разделения передающей среды в свои спецификации наряду с режимом работы по индивидуальным линиям связи.

При использовании индивидуальных линий связи в полносвязных топологиях конечные узлы должны иметь по одному порту на каждую линию связи. В звездообразных топологиях конечные узлы могут подключаться индивидуальными линиями связи к специальному устройству – коммутатору. В глобальных сетях коммутаторы использовались уже на начальном этапе, а в локальных сетях – с начала 90-х годов. Необходимо подчеркнуть, что индивидуальными в таких сетях являются только линии связи между конечными узлами и коммутаторами сети, а связи между коммутаторами остаются разделяемыми, так как по ним передаются сообщения разных конечных узлов (рисунок 1.9).

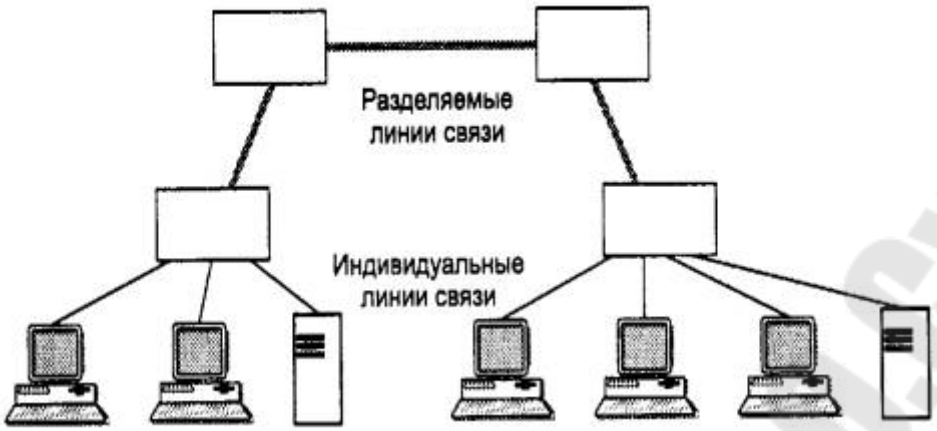


Рисунок 1.9– Индивидуальные и разделяемые линии связи в сетях на основе коммутаторов

В глобальных сетях отказ от разделяемых линий связи объясняется техническими причинами. Здесь большие временные задержки распространения сигналов принципиально ограничивают применимость техники разделения линии связи. Компьютеры могут затратить больше времени на переговоры о том, кому сейчас можно использовать линию связи, чем непосредственно на передачу данных по этой линии связи. Однако это не относится к линиям связи типа «коммутатор – коммутатор». В этом случае только два коммутатора борются за доступ к линии связи, и это существенно упрощает задачу организации совместного использования линии.

Адресация компьютеров

Еще одной новой проблемой, которую нужно учитывать при объединении трех и более компьютеров, является проблема их адресации. К адресу узла сети и схеме его назначения можно предъявить несколько требований.

- Адрес должен уникально идентифицировать компьютер в сети любого масштаба.
- Схема назначения адресов должна сводить к минимуму ручной труд администратора и вероятность дублирования адресов.
- Адрес должен иметь иерархическую структуру, удобную для построения больших сетей. Эту проблему хорошо иллюстрируют международные почтовые адреса, которые позволяют почтовой службе, организующей доставку писем между странами, пользоваться только названием страны адресата и не учитывать название его города, а тем более улицы. В больших сетях, состоящих из многих тысяч узлов, отсутствие иерархии адреса может привести к большим

издержкам – конечным узлам и коммуникационному оборудованию придется оперировать с таблицами адресов, состоящими из тысяч записей.

- Адрес должен быть удобен для пользователей сети, а это значит, что он должен иметь символьное представление например, Servers или www.cisco.com.
- Адрес должен иметь по возможности компактное представление, чтобы не перегружать память коммуникационной аппаратуры – сетевых адаптеров, маршрутизаторов и т. п.

Нетрудно заметить, что эти требования противоречивы – например, адрес, имеющий иерархическую структуру, скорее всего будет менее компактным, чем неиерархический (такой адрес часто называют «плоским», то есть не имеющим структуры). Символьный же адрес скорее всего потребует больше памяти, чем адрес-число.

Так как все перечисленные требования трудно совместить в рамках какой-либо одной схемы адресации, то на практике обычно используется сразу несколько схем, так что компьютер одновременно имеет несколько адресов-имен. Каждый адрес используется в той ситуации, когда соответствующий вид адресации наиболее удобен. А чтобы не возникало путаницы и компьютер всегда однозначно определялся своим адресом, используются специальные вспомогательные протоколы, которые по адресу одного типа могут определить адреса других типов.

Наибольшее распространение получили три схемы адресации узлов.

- *Аппаратные (hardware) адреса*. Эти адреса предназначены для сети небольшого или среднего размера, поэтому они не имеют иерархической структуры. Типичным представителем адреса такого типа является адрес сетевого адаптера локальной сети. Такой адрес обычно используется только аппаратурой, поэтому его стараются сделать по возможности компактным и записывают в виде двоичного или шестнадцатеричного значения, например 0081005e24a8. При задании аппаратных адресов обычно не требуется выполнение ручной работы, так как они либо встраиваются в аппаратуру компанией-изготовителем, либо генерируются автоматически при каждом новом запуске оборудования, причем уникальность адреса в пределах сети обеспечивает оборудование. Помимо отсутствия иерархии, использование аппаратных адресов связано еще с одним недостатком – при замене аппаратуры, например, сетевого адаптера, изменяется и

адрес компьютера. Более того, при установке нескольких сетевых адаптеров у компьютера появляется несколько адресов, что не очень удобно для пользователей сети.

- *Символьные адреса* или *имена*. Эти адреса предназначены для запоминания людьми и поэтому обычно несут смысловую нагрузку. Символьные адреса легко использовать как в небольших, так и крупных сетях. Для работы в больших сетях символьное имя может иметь сложную иерархическую структуру, например ftp-archl.ucl.ac.uk. Этот адрес говорит о том, что данный компьютер поддерживает ftp-архив в сети о дного из колледжей Лондонского университета (UniversityCollegeLondon – ucl) и эта сеть относится к академической ветви (ac) Internet Великобритании (UnitedKingdom – uk). При работе в пределах сети Лондонского университета такое длинное символьное имя явно избыточно и вместо него удобно пользоваться кратким символьным именем, на роль которого хорошо подходит самая младшая составляющего полного имени, то есть имя ftp-archl.
- Числовые составные адреса. Символьные имена удобны для людей, но из-за переменного формата и потенциально большой длины их передача по сети не очень экономична. Поэтому во многих случаях для работы в больших сетях в качестве адресов узлов используют числовые составные адреса фиксированного и компактного форматов. Типичным представителями адресов этого типа являются IP- и IPX-адреса. В них поддерживается двухуровневая иерархия, адрес делится на старшую часть – номер сети и младшую – номер узла. Такое деление позволяет передавать сообщения между сетями только на основании номера сети, а номер узла используется только после доставки сообщения в нужную сеть; точно так же, как название улицы используется почтальоном только после того, как письмо доставлено в нужный город. В последнее время, чтобы сделать маршрутизацию в крупных сетях более эффективной, предлагаются более сложные варианты числовой адресации, в соответствии с которыми адрес имеет три и более составляющих. Такой подход, в частности, реализован в новой версии протокола IPv6, предназначенного для работы в сети Internet. В современных сетях для адресации узлов применяются, как правило, одновременно все три приведенные выше схемы. Пользователи адресуют компьютеры символьными именами, которые автоматически заменяются в сообщениях, передаваемых по сети, на числовые

номера. С помощью этих числовых номеров сообщения передаются из одной сети в другую, а после доставки сообщения в сеть назначения вместо числового номера используется аппаратный адрес компьютера. Сегодня такая схема характерна даже для небольших автономных сетей, где, казалось бы, она явно избыточна – это делается для того, чтобы при включении этой сети в большую сеть не нужно было менять состав операционной системы.

Проблема установления соответствия между адресами различных типов, которой занимается *служба разрешения имен*, может решаться как полностью централизованными, так и распределенными средствами. В случае централизованного подхода в сети выделяется один компьютер (сервер имен), в котором хранится таблица соответствия друг другу имен различных типов, например символьных имен и числовых номеров. Все остальные компьютеры обращаются к серверу имен, чтобы по символьному имени найти числовой номер компьютера, с которым необходимо обменяться данными.

При другом, распределенном подходе, каждый компьютер сам решает задачу установления соответствия между именами. Например, если пользователь указал для узла назначения числовой номер, то перед началом передачи данных компьютер-отправитель посылает всем компьютерам сети сообщение (такое сообщение называется широковещательным) с просьбой опознать это числовое имя. Все компьютеры, получив это сообщение, сравнивают заданный номер со своим собственным. Тот компьютер, у которого обнаружилось совпадение, посылает ответ, содержащий его аппаратный адрес, после чего становится возможным отправка сообщений по локальной сети.

Распределенный подход хорош тем, что не предполагает выделения специального компьютера, который к тому же часто требует ручного задания таблицы соответствия имен. Недостатком распределенного подхода является необходимость широковещательных сообщений – такие сообщения перегружают сеть, так как они требуют обязательной обработки всеми узлами, а не только узлом назначения. Поэтому распределенный подход используется только в небольших локальных сетях. В крупных сетях распространение широковещательных сообщений по всем ее сегментам становится практически нереальным, поэтому для них характерен централизованный подход. Наиболее известной службой

централизованного разрешения имен является служба DomainNameSystem (DNS) сети Internet.

1.5.6Ethernet – пример стандартной локальной сети

Рассмотрим, каким образом описанные выше общие подходы к решению наиболее важных проблем построения сетей воплощены в наиболее популярной сетевой технологии – *Ethernet*.

Сетевая технология– это согласованный набор стандартных протоколов и реализующих их программно-аппаратных средств (например, сетевых адаптеров, драйверов, кабелей и разъемов), достаточный для построения компьютерной сети. Эпитет «достаточный» подчеркивает то обстоятельство, что этот набор представляет собой минимальный набор средств, с помощью которых можно построить работоспособную сеть. Возможно, эту сеть можно улучшить, например, за счет выделения в ней подсетей, что сразу потребует кроме протоколов стандарта Ethernet применения протокола IP, а также специальных коммуникационных устройств – маршрутизаторов. Улучшенная сеть будет, скорее всего, более надежной и быстродействующей, но за счет надстроек над средствами технологии Ethernet, которая составила базис сети.

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

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

Стандарт Ethernet был принят в 1980 году. Основной принцип, положенный в основу Ethernet, – *случайный метод доступа* к разделяемой среде передачи данных. В качестве такой среды может использоваться толстый или тонкий коаксиальный кабель, витая пара, оптоволокно или радиоволны (кстати, первой сетью, построенной на принципе случайного доступа к разделяемой среде, была радиосеть Aloha Гавайского университета).

В стандарте Ethernet строго зафиксирована топология электрических связей. Компьютеры подключаются к разделяемой среде в соответствии с типовой структурой «общая шина» (рисунок 1.10). С помощью разделяемой во времени шины любые два компьютера могут обмениваться данными. Управление доступом к линии связи осуществляется специальными контроллерами – сетевыми адаптерами Ethernet. Каждый компьютер, а более точно, каждый сетевой адаптер, имеет уникальный адрес. Передача данных происходит со скоростью 10 Мбит/с. Эта величина является пропускной способностью сети Ethernet.

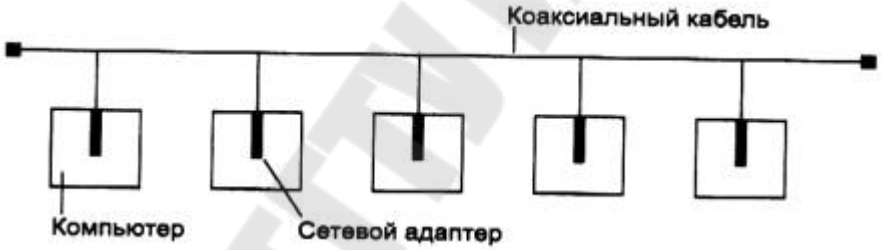


Рисунок 1.10– Сеть Ethernet

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

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

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

Иногда может возникать ситуация, когда одновременно два или более компьютера решают, что сеть свободна, и начинают передавать информацию. Такая ситуация, называемая *коллизией*, препятствует правильной передаче данных по сети. В стандарте Ethernet предусмотрен алгоритм обнаружения и корректной обработки коллизий. Вероятность возникновения коллизии зависит от интенсивности сетевого трафика.

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

Главным достоинством сетей Ethernet, благодаря которому они стали такими популярными, является их экономичность. Для построения сети достаточно иметь по одному сетевому адаптеру для каждого компьютера плюс один физический сегмент коаксиального кабеля нужной длины. Другие базовые технологии, например TokenRing, для создания даже небольшой сети требуют наличия дополнительного устройства – концентратора.

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

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

Другие базовые сетевые технологии – TokenRing, FDDI, l00VGAny-LAN, хотя и обладают многими индивидуальными чертами, в то же время имеют много общих свойств с Ethernet. В первую очередь – это применение регулярных фиксированных топологий (иерархическая звезда и кольцо), а также разделяемых сред передачи данных. Существенные отличия одной технологии от

другой связаны с особенностями используемого метода доступа к разделяемой среде. Так, отличия технологии Ethernet от технологии TokenRing во многом определяются спецификой заложенных в них методов разделения среды – случайного алгоритма доступа в Ethernet и метода доступа путем передачи маркера в TokenRing.

1.5.7 Структуризация как средство построения больших сетей

В сетях с небольшим (10-30) количеством компьютеров чаще всего используется одна из типовых топологий – общая шина, кольцо, звезда или полносвязная сеть. Все перечисленные топологии обладают свойством однородности, то есть все компьютеры в такой сети имеют одинаковые права в отношении доступа к другим компьютерам (за исключением центрального компьютера при соединении звезда). Такая однородность структуры делает простой процедуру наращивания числа компьютеров, облегчает обслуживание и эксплуатацию сети.

Однако при построении больших сетей однородная структура связей превращается из преимущества в недостаток. В таких сетях использование типовых структур порождает различные ограничения, важнейшими из которых являются:

- ограничения на длину связи между узлами;
- ограничения на количество узлов в сети;
- ограничения на интенсивность трафика, порождаемого узлами сети.

Например, технология Ethernet на тонком коаксиальном кабеле позволяет использовать кабель длиной не более 185 метров, к которому можно подключить не более 30 компьютеров. Однако, если компьютеры интенсивно обмениваются информацией между собой, иногда приходится снижать число подключенных к кабелю компьютеров до 20, а то и до 10, чтобы каждому компьютеру доставалась приемлемая доля общей пропускной способности сети.

Для снятия этих ограничений используются специальные методы структуризации сети и специальное структурообразующее оборудование – повторители, концентраторы, мосты, коммутаторы, маршрутизаторы. Оборудование такого рода также называют коммуникационным, имея в виду, что с помощью него отдельные сегменты сети взаимодействуют между собой.

Физическая структуризация сети

Простейшее из коммуникационных устройств – *повторитель (repeater)* – используется для физического соединения различных сегментов кабеля локальной сети с целью увеличения общей длины сети. Повторитель передает сигналы, приходящие из одного сегмента сети, в другие ее сегменты (рисунок 1.11). Повторитель позволяет преодолеть ограничения на длину линий связи за счет улучшения качества передаваемого сигнала – восстановления его мощности и амплитуды, улучшения фронтов и т. п.

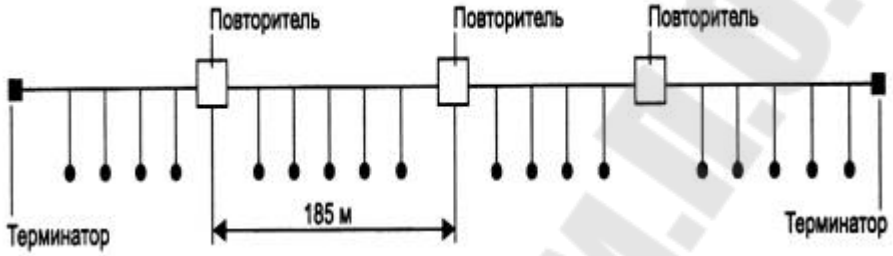


Рисунок 1.11– Повторитель позволяет увеличить длину сети Ethernet

Концентраторы характерны практически для всех базовых технологий локальных сетей – Ethernet, ArcNet, TokenRing, FDDI, FastEthernet, GigabitEthernet, l00VG-AnyLAN.

Нужно подчеркнуть, что в работе концентраторов любых технологий много общего – они повторяют сигналы, пришедшие с одного из своих портов, на других своих портах. Разница состоит в том, на каких именно портах повторяются входные сигналы. Так, концентратор Ethernet повторяет входные сигналы на всех своих портах, кроме того, с которого сигналы поступают (рисунок 1.12, а). А концентратор TokenRing (рисунок 1.12, б) повторяет входные сигналы, поступающие с некоторого порта, только на одном порту – на том, к которому подключен следующий в кольце компьютер.

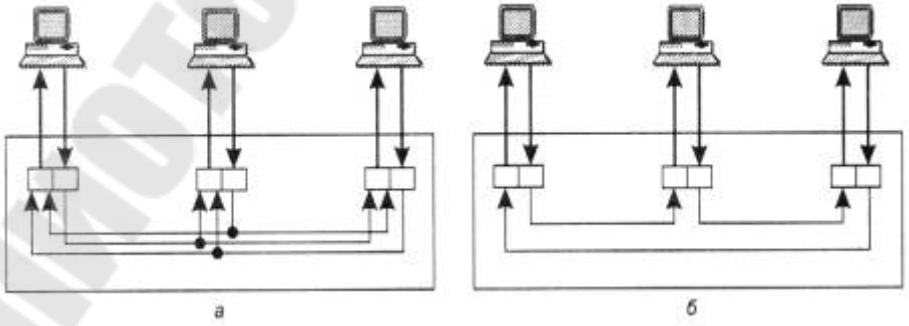


Рисунок 1.12– Концентраторы различных технологий

Обратим внимание, что концентратор всегда изменяет физическую топологию сети, но при этом оставляет без изменения ее логическую топологию.Напомним, что под физической топологией понимается конфигурация связей, образованных отдельными частями кабеля, а под логической – конфигурация информационных потоков между компьютерами сети. Во многих случаях физическая и логическая топологии сети совпадают. Например, сеть, представленная на рисунке 1.13, а, имеет физическую топологию кольцо. Компьютеры этой сети получают доступ к кабелям кольца за счет передачи друг другу специального кадра – маркера, причем этот маркер также передается последовательно от компьютера к компьютеру в том же порядке, в котором компьютеры образуют физическое кольцо, то есть компьютер А передает маркер компьютеру В, компьютер В – компьютеру С и т. д.

Сеть, показанная на рисунке 1.13, б, демонстрирует пример несовпадения физической и логической топологии. Физически компьютеры соединены по топологии общая шина. Доступ же к шине происходит не по алгоритму случайного доступа, применяемому в технологии Ethernet, а путем передачи маркера в кольцевом порядке: от компьютера А – компьютеру В, от компьютера В – компьютеру С и т. д. Здесь порядок передачи маркера уже не повторяет физические связи, а определяется логическим конфигурированием драйверов сетевых адаптеров. Ничто не мешает настроить сетевые адаптеры и их драйверы так, чтобы компьютеры образовали кольцо в другом порядке, например: В, А, С... При этом физическая структура сети никак не изменяется.

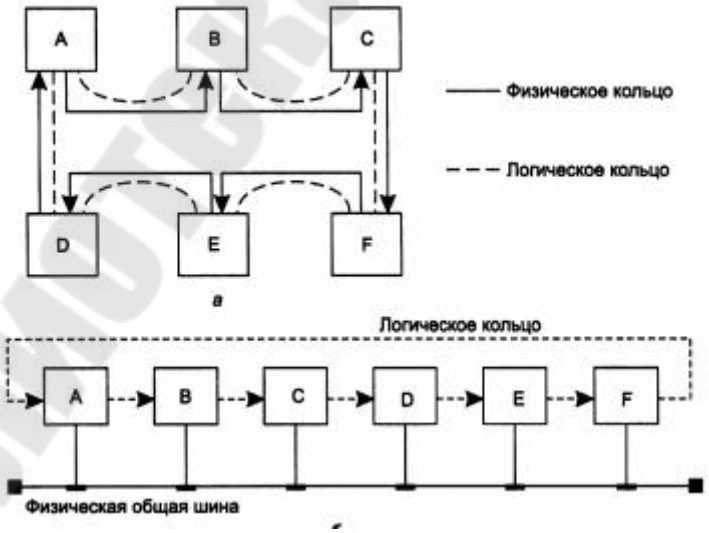


Рисунок 1.13– Логическая и физическая топологии сети

Другим примером несовпадения физической и логической топологий сети является уже рассмотренная сеть на рисунке 1.12, а. Концентратор Ethernet поддерживает в сети физическую топологию звезда. Однако логическая топология сети осталась без изменений – это общая шина. Так как концентратор повторяет данные, пришедшие с любого порта, на всех остальных портах, то они появляются одновременно на всех физических сегментах сети, как и в сети с физической общей шиной. Логика доступа к сети совершенно не меняется: все компоненты алгоритма случайного доступа – определение незанятости среды, захват среды, распознавание и отработка коллизий – остаются в силе.

Физическая структуризация сети с помощью концентраторов полезна не только для увеличения расстояния между узлами сети, но и для повышения ее надежности. Например, если какой-либо компьютер сети Ethernet с физической общей шиной из-за сбоя начинает непрерывно передавать данные по общему кабелю, то вся сеть выходит из строя, и для решения этой проблемы остается только один выход – вручную отсоединить сетевой адаптер этого компьютера от кабеля. В сети Ethernet, построенной с использованием концентратора, эта проблема может быть решена автоматически – концентратор отключает свой порт, если обнаруживает, что присоединенный к нему узел слишком долго монопольно занимает сеть. Концентратор может блокировать некорректно работающий узел и в других случаях, выполняя роль некоторого управляющего узла.

Логическая структуризация сети

Физическая структуризация сети полезна во многих отношениях, однако в ряде случаев, обычно относящихся к сетям большого и среднего размера, невозможно обойтись без логической структуризации сети. Наиболее важной проблемой, не решаемой путем физической структуризации, остается проблема перераспределения передаваемого трафика между различными физическими сегментами сети.

В большой сети естественным образом возникает неоднородность информационных потоков: сеть состоит из множества подсетей рабочих групп, отделов, филиалов предприятия и других административных образований. Очень часто наиболее интенсивный обмен данными наблюдается между компьютерами, принадлежащими к одной подсети, и только небольшая часть

обращений происходит к ресурсам компьютеров, находящихся вне локальных рабочих групп. (До недавнего времени такое соотношение трафиков не подвергалось сомнению, и был даже сформулирован эмпирический закон «80/20», в соответствии с которым в каждой подсети 80 % трафика является внутренним и только 20 % – внешним.) Сейчас характер нагрузки сетей во многом изменился, широко внедряется технология intranet, на многих предприятиях имеются централизованные хранилища корпоративных данных, активно используемые всеми сотрудниками предприятия. Все это не могло не повлиять на распределение информационных потоков. И теперь не редки ситуации, когда интенсивность внешних обращений выше интенсивности обмена между «соседними» машинами. Но независимо от того, в какой пропорции распределяются внешний и внутренний трафик, для повышения эффективности работы сети неоднородность информационных потоков необходимо учитывать.

Сеть с типовой топологией (шина, кольцо, звезда), в которой все физические сегменты рассматриваются в качестве одной разделяемой среды, оказывается неадекватной структуре информационных потоков в большой сети. Например, в сети с общей шиной взаимодействие любой пары компьютеров занимает ее на все время обмена, поэтому при увеличении числа компьютеров в сети шина становится узким местом. Компьютеры одного отдела вынуждены ждать, когда окончит обмен пара компьютеров другого отдела, и это при том, что необходимость в связи между компьютерами двух разных отделов возникает гораздо реже и требует совсем небольшой пропускной способности.

Этот случай иллюстрирует рисунок 1.14, а. Здесь показана сеть, построенная с использованием концентраторов. Пусть компьютер А, находящийся в одной подсети с компьютером В, посылает ему данные. Несмотря на разветвленную физическую структуру сети, концентраторы распространяют любой кадр по всем ее сегментам. Поэтому кадр, посылаемый компьютером А компьютеру В, хотя и не нужен компьютерам отделов 2 и 3, в соответствии с логикой работы концентраторов поступает на эти сегменты тоже. И до тех пор, пока компьютер В не получит адресованный ему кадр, ни один из компьютеров этой сети не сможет передавать данные.

Такая ситуация возникает из-за того, что логическая структура данной сети осталась однородной – она никак не учитывает увеличение интенсивности трафика внутри отдела и предоставляет

всем парам компьютеров равные возможности по обмену информацией (рисунок 1.14, б).

Решение проблемы состоит в отказе от идеи единой однородной разделяемой среды. Например, в рассмотренном выше примере желательно было бы сделать так, чтобы кадры, которые передают компьютеры отдела 1, выходили бы за пределы этой части сети в том и только в том случае, если эти кадры направлены какому-либо компьютеру из других отделов. С другой стороны, в сеть каждого из отделов должны попадать те и только те кадры, которые адресованы узлам этой сети. При такой организации работы сети ее производительность существенно повыситься, так как компьютеры одного отдела не будут простаивать в то время, когда обмениваются данными компьютеры других отделов.

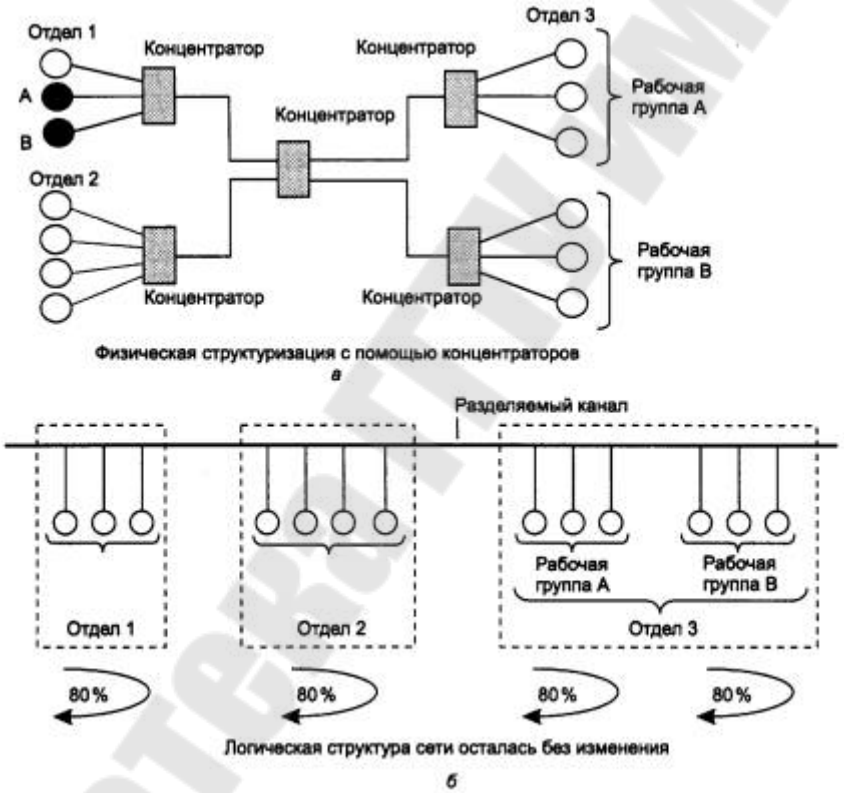


Рисунок 1.14– Противоречие между логической структурой сети и структурой информационных потоков

Нетрудно заметить, что в предложенном решении мы отказались от идеи общей разделяемой среды в пределах всей сети, хотя и оставили ее в пределах каждого отдела. Пропускная способность линий связи между отделами не должна совпадать с пропускной способностью среды внутри отделов. Если трафик между

отделами составляет только 20 % трафика внутри отдела (как уже отмечалось, эта величина может быть другой), то и пропускная способность линий связи и коммуникационного оборудования, соединяющего отделы, может быть значительно ниже внутреннего трафика сети отдела.

Обратим внимание, что распространение трафика, предназначенного для компьютеров некоторого сегмента сети, только в пределах этого сегмента, называется *локализацией* трафика. *Логическая структуризация сети* – это процесс разбиения сети на сегменты с локализованным трафиком.

Для логической структуризации сети используются такие коммуникационные устройства, как мосты, коммутаторы, маршрутизаторы и шлюзы.

Мост (bridge) делит разделяемую среду передачи сети на части (часто называемые логическими сегментами), передавая информацию из одного сегмента в другой только в том случае, если такая передача действительно необходима, то есть если адрес компьютера назначения принадлежит другой подсети. Тем самым мост изолирует трафик одной подсети от трафика другой, повышая общую производительность передачи данных в сети. Локализация трафика не только экономит пропускную способность, но и уменьшает возможность несанкционированного доступа к данным, так как кадры не выходят за пределы своего сегмента и их сложнее перехватить злоумышленнику.

На рисунок 1.15 показана сеть, которая была получена из сети с центральным концентратором (см. рисунок 1.14) путем его замены на мост. Сети 1-го и 2-го отделов состоят из отдельных логических сегментов, а сеть отдела 3 – из двух логических сегментов. Каждый логический сегмент построен на базе концентратора и имеет простейшую физическую структуру, образованную отрезками кабеля, связывающими компьютеры с портами концентратора.

Мосты используют для локализации трафика аппаратные адреса компьютеров. Это затрудняет распознавание принадлежности того или иного компьютера к определенному логическому сегменту – сам адрес не содержит никакой информации по этому поводу. Поэтому мост достаточно упрощенно представляет деление сети на сегменты – он запоминает, через какой порт на него поступил кадр данных от каждого компьютера сети, и в дальнейшем передает кадры, предназначенные для этого компьютера, на этот порт. Точной

топологии связей между логическими сегментами мост не знает. Из-за этого применение мостов приводит к значительным ограничениям на конфигурацию связей сети – сегменты должны быть соединены таким образом, чтобы в сети не образовывались замкнутые контуры.

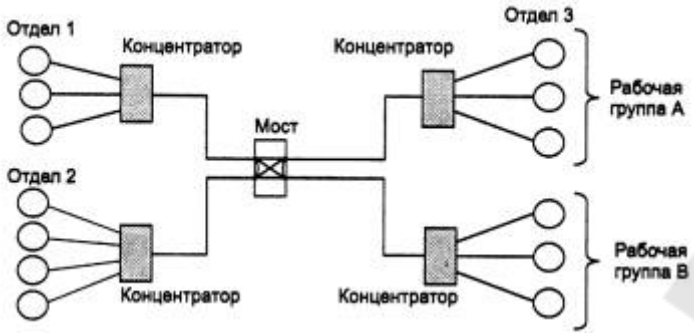


Рисунок 1.15– Логическая структуризация сети с помощью моста

Коммутатор (switch, switchinghub) по принципу обработки кадров ничем не отличается от моста. Основное его отличие от моста состоит в том, что он является своего рода коммуникационным мультипроцессором, так как каждый его порт оснащен специализированным процессором, который обрабатывает кадры по алгоритму моста независимо от процессоров других портов. За счет этого общая производительность коммутатора обычно намного выше производительности традиционного моста, имеющего один процессорный блок. Можно сказать, что коммутаторы – это мосты нового поколения, которые обрабатывают кадры в параллельном режиме.

Ограничения, связанные с применением мостов и коммутаторов – по топологии связей, а также ряд других, – привели к тому, что в ряду коммуникационных устройств появился еще один тип оборудования – *маршрутизатор (router)*. Маршрутизаторы более надежно и более эффективно, чем мосты, изолируют трафик отдельных частей сети друг от друга. Маршрутизаторы образуют логические сегменты посредством явной адресации, поскольку используют не плоские аппаратные, а составные числовые адреса. В этих адресах имеется поле номера сети, так что все компьютеры, у которых значение этого поля одинаково, принадлежат к одному сегменту, называемому в данном случае *подсетью (subnet)*.

Кроме локализации трафика маршрутизаторы выполняют еще много других полезных функций. Так, маршрутизаторы могут

работать в сети с замкнутыми контурами, при этом они осуществляют выбор наиболее рационального маршрута из нескольких возможных. Сеть, представленная на рисунке 1.16, отличается от своей предшественницы (см. рисунок 1.15) тем, что между подсетями отделов 1 и 2 проложена дополнительная связь, которая может использоваться как для повышения производительности сети, так и для повышения ее надежности.

Рисунок 1.16 – Логическая структуризация сети с помощью маршрутизаторов

Другой очень важной функцией маршрутизаторов является их способность связывать в единую сеть подсети, построенные с использованием разных сетевых технологий, например Ethernet и X.25.

Кроме перечисленных устройств отдельные части сети может соединять *шлюз (gateway)*. Обычно основной причиной, по которой в сети используют шлюз, является необходимость объединить сети с разными типами системного и прикладного программного обеспечения, а не желание локализовать трафик. Тем не менее, шлюз обеспечивает и локализацию трафика в качестве некоторого побочного эффекта.

Крупные сети практически никогда не строятся без логической структуризации. Для отдельных сегментов и подсетей характерны типовые однородные топологии базовых технологий, и для их объединения всегда используется оборудование, обеспечивающее локализацию трафика, – мосты, коммутаторы, маршрутизаторы и шлюзы.

1.5.8 Маршрутизация и маршрутизаторы

Протоколы сетевого уровня реализуются, как правило, в виде программных модулей и выполняются на конечных узлах – компьютерах, называемых *хостами,* а также на промежуточных узлах – маршрутизаторах, называемых *шлюзами.* М*аршрутизаторы*

*(router)*может представлять собой как специализированное устройство, так и универсальный компьютер.

Компьютерная сеть в общем случае рассматривается как совокупность нескольких сетей (Рисунок1.17).

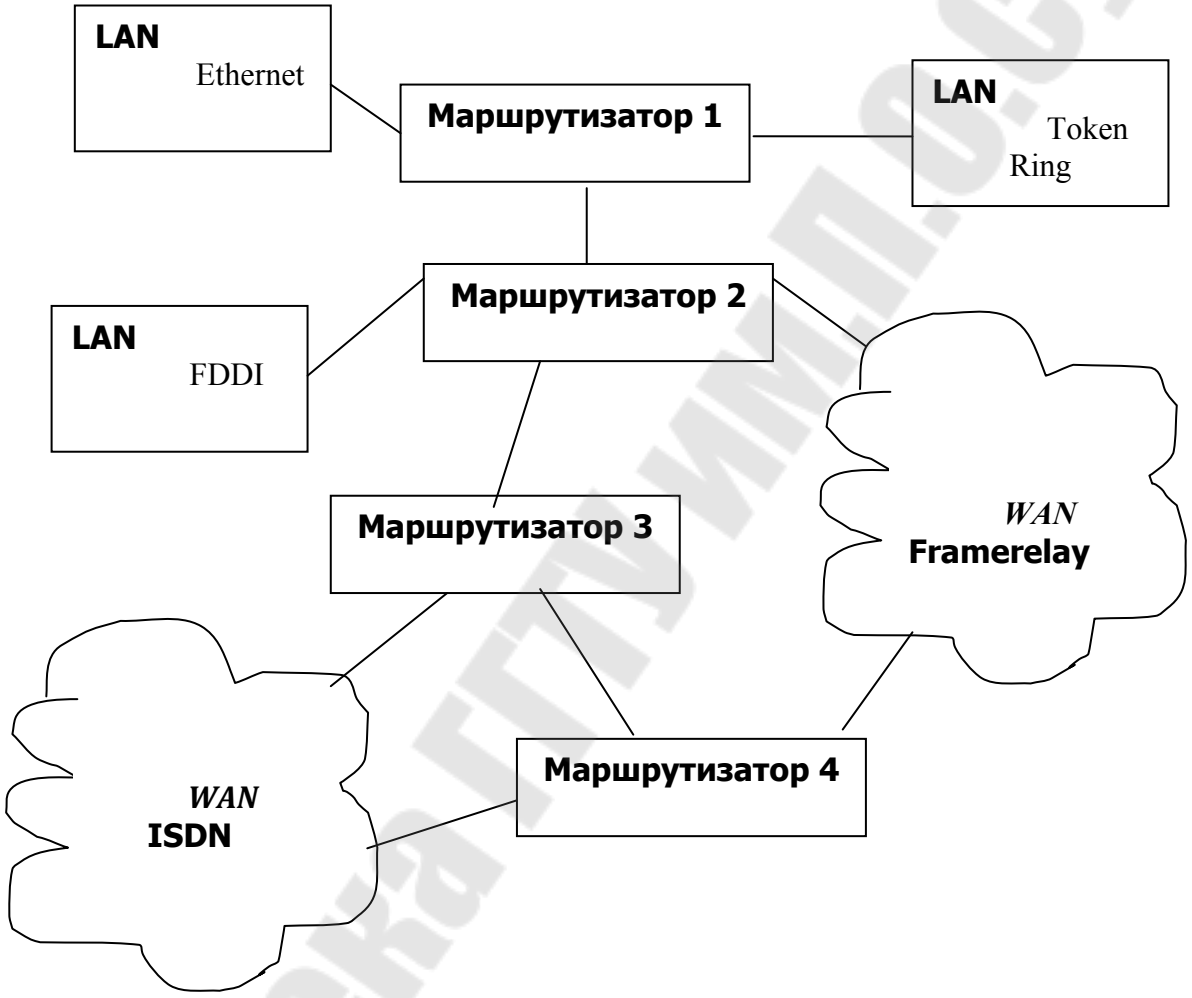


Рисунок1.17– Пример компьютерной сети в общем случае

Сети, входящие в основную сеть называются *подсетями или просто сетями.* Подсетями могут быть как локальные, так и глобальные компьютерные сети. Следует отметить, что внутри одной подсети на канальном уровне используется единая технология из рассмотренных в предыдущих лекциях..

Маршрутизаторы связывают подсети между собой путем непосредственной адресации каждой из подсетей. Способом формирования сетевого адреса является уникальная нумерация всех

подсетей составной сети и нумерация всех узлов в пределах каждой подсети. Таким образом, сетевой адрес представляет собой пару: *номер сети (подсети) и номер узла*. Номер узла называется также локальным адресом.

Продвижение пакетов между подсетями, в соответствии с адресами назначения называется *маршрутизацией*. На Рисунке 1.18 показан принцип работы маршрутизатора, где в качестве примера приведена таблица маршрутизации маршрутизатора М4.

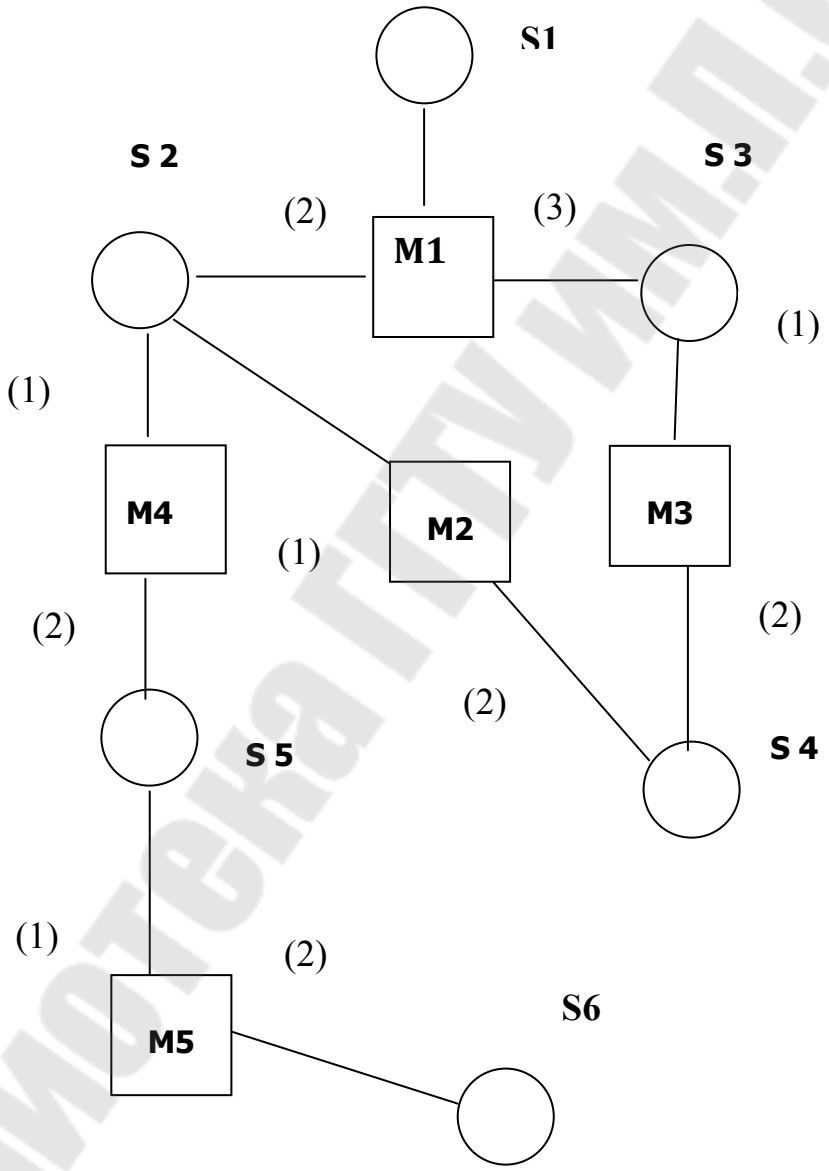


Рисунок1.18 – Принцип работы маршрутизатора

Таблица 1.1– Таблица маршрутизатора М4

Сеть	Сетевой адрес следующего маршрутизатора	Сетевой адрес выходного порта	Расстояние до сети назначения
S1	М1 (2)	М4 (1)	1
S2	-	М4 (1)	0
S3	М1 (2)	М4 (1)	1
S4	М2 (1)	М4 (1)	1
S5	-	М4 (2)	0
S6	М5 (1)	М 4(2)	1

Для выбора маршрута пересылки пакета маршрутизатор анализирует специальную *таблицу маршрутизации*, которая хранится в памяти маршрутизатора (см. таблицу 1.1). В первом столбце таблицы перечислены номера сетей, входящих в общую сеть. В каждой строке таблицы следом за номером сети указывается сетевой адрес следующего маршрутизатора (точнее его соответствующий порт), на который надо направить пакет, чтобы тот передвигался к сети с данным номером. Когда на маршрутизатор поступает новый пакет, из него извлекается номер сети назначения, сравнивается с каждой строкой таблицы. В случае совпадения номера, записанного в таблицу, с требуемым номером пакет будет передвигаться в этом направлении.

Если, например, маршрутизатор 4 принял пакет, который предназначен для сети S6, то в последней строке своей таблицы маршрутизации он определит, что сеть S6 подключена к первому порту маршрутизатора М5, а для того, чтобы пакет попал на маршрутизатор 5, маршрутизатор 4 должен отправить пакет на свой второй порт.

Таблицы маршрутизации могут составляться либо *статически,* либо *динамически.* Статические таблицы прописываются вручную администратором сети, а динамические таблицы строятся самими маршрутизаторами, которые обмениваются между собой информацией о конфигурации сети с помощью специальных служебных протоколов (например, RIP, OSPF, NLSP).

При использовании динамических алгоритмов таблица маршрутизации автоматически обновляется при изменении топологии

сети или интенсивности информационных потоков (трафика). Реализуемые специальными служебными протоколами алгоритмы определения состояния сети различаются по способу получения информации, времени изменения маршрутов и используемым показателям оценки того или иного маршрута.

Одно-маршрутные алгоритмы определяют только один маршрут, при этом он может оказаться не оптимальным. Многомаршрутные алгоритмы предлагают несколько маршрутов к одному и тому же получателю. Такие алгоритмы позволяют передавать пакеты получателю по нескольким каналам связи одновременно, что повышает пропускную способность и надежность сети в целом.

Кроме основных функций маршрутизации на сетевом уровне маршрутизатором осуществляется фильтрация передаваемой по сети информации. Маршрутизаторы по сравнению с концентраторами и коммутаторами являются более «интеллектуальными» устройствами. С помощью своего программного обеспечения маршрутизаторы способны производить анализ отдельных полей в кадре, администратор сети может с их помощью задавать сложные алгоритмы фильтрации данных. Они, например, могут запретить прохождение в сети всех пакетов, кроме пакетов сети, принадлежащей конкретному предприятию. Маршрутизаторы могут анализировать структуру сообщений верхнего уровня модели OSI, не пропускать в сеть сообщения определенных служб (например, FPT).

Они могут реализовывать алгоритмы обслуживания очередей данных. Связь маршрутизатора (т.е. сетевого уровня) с канальным осуществляется с помощью преобразования сетевого адреса в локальный адрес той сети, где используется определенная технология канального и физического уровня. Для этого сетевой протокол обращается к *протоколу разрешения адресов.(ARP)* Этот протокол устанавливает соответствие между сетевыми и локальными адресами либо на основании заранее составленных таблиц, либо рассылкой широковещательных запросов, которые определяют и возвращают машрутизатору локальные адреса.

Пакет сетевого уровня снабжается сетевым заголовком, в котором указываются длина пакета, адрес источника, адрес получателя, время жизни пакета и другая служебная информация. С сетевого уровня пакет, локальный адрес следующего маршрутизатора, а также номер порта маршрутизатора отправителя

предаются вниз, канальному уровню.. На основании указанного номера порта выполняется упаковка пакета в кадр соответствующего формата, предписываемого соответствующей технологией канального уровня. Другими словами, маршрутизатор заранее «знает», к какому порту подключена подсеть, и какая именно технология канального уровня реализуется в этой подсети. В поле адреса назначения заголовка кадра помещается локальный адрес следующего маршрутизатора. Если маршрутизатор реализован на персональном компьютере, то локальный адрес канального уровня представляет собой физический адрес сетевой платы компьютера (напомним, что он задается заводом изготовителем и называется MAC – адресом). Сформированный таким образом кадр отправляется в сеть.

Для работы на сетевом уровне используются различные протоколы, такие как TCP\IP и IPX/ SPX, причем в последнее время протокол TCP/ IP вышел в абсолютные лидеры. Более подробно протокол TCP/IP будет рассмотрен ниже в теме, посвященной изучению Интернет.

Как отмечалось выше, современные маршрутизаторы в состоянии находить наилучший путь продвижения пакетов, т.е. в некотором смысле оптимизировать маршрутизацию. Маршрутизаторы могут поддерживать как один протокол сетевого уровня (например, IP), так и множество протоколов. В последнем случае такие маршрутизаторы называются *многопротокольными.*

По областям применения маршрутизаторы можно классифицировать:

- *магистральные маршрутизаторы*. Они предназначены для построения центральной сети фирмы или предприятия. Такая сеть, как правило, состоит из большого числа локальных сетей, расположенных в разных зданиях и даже регионах, использующих разнообразные технологии канального уровня. Магистральные маршрутизаторы- это наиболее мощные устройства, способные обрабатывать несколько тысяч или даже несколько миллионов пакетов в секунду.
- *региональные маршрутизаторы* соединяют региональные отделения фирмы или предприятия между собой и центральной сетью. Маршрутизаторы этого типа представляют собой некоторую упрощенную версию магистральных маршрутизаторов. Это наиболее обширный класс маршрутизаторов, из имеющихся на рынке.

- *маршрутизаторы удаленных офисов* соединяют, как правило, локальную сети удаленного офиса с центральной сетью или с сетью регионального отделения.
- *маршрутизаторы локальных сетей* предназначены для разделения крупных локальных сетей на подсети. Основное требование, предъявляемое к таким маршрутизаторам – это высокая скорость маршрутизации. Все порты работают на скорости 10 – 100 Мбит/ с.

На современном рынке сетевого оборудования лидерство по маршрутизаторам держат фирмы Cisco и 3Com.

1.5.9 Тенденции развития маршрутизаторов

Традиционное построение компьютерной сети с использованием концентраторов, коммутаторов и маршрутизаторов выглядит таким образом, что на нижнем уровне располагаются сегменты сети, быстро работающие на концентраторах и коммутаторах. На более высоком уровне располагается маршрутизатор, к которому подключено определенное количество локальных сетей (подсетей). Через порты маршрутизатора проходит информация от компьютеров одной сети к компьютерам другой сети. В общем случае маршрутизатор затрачивает на обработку каждого пакета больше времени, чем коммутатор на обработку кадра, поскольку выполняет более сложную обработку данных, включая алгоритмы фильтрации, выбор оптимального маршрута и т.д.

В настоящее время ситуация в компьютерных сетях быстро меняется. Это в первую очередь связано со стремительным ростом количества пользователей, а также с увеличением числа мультимедийных приложений (аудио, видео и т.д.), которые необходимо передать через сеть. Все это выдвигает требование увеличения производительности сетевого оборудования, что, однако, тормозится низкой скоростью работы маршрутизаторов.

Для разрешения возникшей проблемы могут быть предложены два пути: либо отказаться вообще от маршрутизации, либо увеличить ее производительность.

Первый путь предполагает применять маршрутизацию как можно реже, только там, где от нее никак нельзя отказаться. Например, на границе между локальной и глобальной сетью. С другой стороны, отказ от маршрутизаторов означает отказ от интеллектуальных возможностей обработки пакетов. Это повышает производительность сети, но приводит к потере всех преимуществ,

которые давали маршрутизаторы. Если сравнить маршрутизаторы с коммутаторами, то можно отметить следующее:

- маршрутизаторы более надежно изолируют подсети друг от друга, защищая их от ошибочных кадров, порожденных например вирусами.
- маршрутизаторы обладают более развитыми возможностями защиты сети от несанкционированного доступа за счет реализации функции анализа и фильтрации предаваемой информации
- сеть, не разделенная маршрутизаторами, имеет ограничения на число компьютеров. Например, для самого популярного протокола IP это ограничение составляет 255 компьютеров для сетей самого доступного класса С.

Таким образом, можно сделать вывод, что в сети необходимо сохранять функции маршрутизации в традиционном виде.

Второе направление, предполагающее повышение производительности маршрутизаторов развивается, как это не странно, фирмами производителями коммутаторов. Для этого коммутатор наделяется некоторыми свойствами маршрутизатора, что позволяет достигать скорости маршрутизации в 5-7 миллионов пакетов в секунду. Такие коммутаторы называются *коммутаторами 3-го уровня.*

В указанных коммутаторах функции коммутации и маршрутизации совмещены. Функции коммутации и маршрутизации в одном устройстве могут совмещаться двумя способами: *классическим*, когда маршрутизация выполняется только по пакету, который надо передать между подсетями, а пакет, который адресован своей подсети, коммутируется и *ускоренным*.

При ускоренном способе осуществляется маршрутизация так называемого *устойчивого потока пакетов*.

Устойчивый поток пакетов в первую очередь характерен тем, что, большое количество пакетов имеют один и тот же адрес назначения. В этом случае коммутатор выполняет маршрутизацию только первых пакетов, а остальные пакеты устойчивого потока направляет без анализа вслед первыми пакетами непосредственно по конечному адресу назначения. Такой способ маршрутизации является очень производительным, но имеет те недостатки, что коммутатору необходимо довольно точно определять устойчивый поток, так же направлять пакеты по конечному адресу назначения. Как было

указано выше, в таблице маршрутизации хранятся адреса «ближайших соседей» маршрутизатора и может не быть конечного адреса назначения для устойчивого потока. Для обеспечения ускоренной маршрутизации многими фирмами разрабатываются специальные служебные протоколы, которые позволяют коммутаторам обмениваться между собой информацией о нахождении конечного адреса устойчивого потока.

Сегодня наблюдается четкая тенденция вытеснения традиционных маршрутизаторов высокопроизводительными коммутаторами 3-го уровня, совмещающих в себе функции как коммутации, так и маршрутизации.

Вопросы для самопроверки.

1. Какая информация передается по каналу, связывающему внешние интерфейсы компьютера и периферийного устройства?
2. Какие компоненты включает интерфейс устройства?
3. Какие задачи решает ОС при обмене с периферийным устройством?
4. Какие функции возлагаются на драйвер периферийного устройства?
5. Дайте определение понятия «топология».
6. К какому типу топологии можно отнести структуру, образованную тремя связанными друг с другом узлами (в виде треугольника)?
7. К какому типу топологии можно отнести структуру, образованную четырьмя связанными друг с другом узлами (в виде квадрата)?
8. К какому типу топологии можно отнести структуру, образованную тремя последовательно соединенными друг с другом узлами (последний не связан с первым)?
9. Частным случаем какой топологии является общая шина:
   - полносвязная;
   - кольцо;
   - звезда.
10. Какая из известных топологий обладает повышенной надежностью?

11. Какой тип топологии наиболее распространен сегодня в локальных сетях?

12. Какие требования предъявляются к системе адресации?

13. К какому типу можно отнести следующие адреса:

- www.olifer.net;
- 20-34-a2-00-c2-27;
- 128.145.23.170.

14. Чем неравномерный поток данных отличается от равномерного?

15. Какие параметры передаваемых данных могут служить признаком потока?

16. Какие из утверждений о маршруте, на ваш взгляд, не всегда верны:

- маршрут – это последовательность промежуточных узлов (интерфейсов), которые проходят данные по пути от отправителя к получателю;
- при определении маршрута всегда выбирается один из нескольких возможных путей;
- каждый маршрут назначается для определенного потока данных;
- из нескольких возможных маршрутов всегда выбирается оптимальный.

17. Опишите основные подходы и критерии, используемые при выборе маршрута.

18. Какие из этих утверждений могут быть в некоторых случаях верными:

- маршруты фиксируются в коммутаторах путем жесткого соединения пар интерфейсов;
- маршруты определяются администратором и заносятся вручную в специальную таблицу;
- таблица маршрутов строится автоматически сетевым программно-аппаратным обеспечением;
- для каждого коммутатора строится своя таблица маршрутов, которая на нем и хранится.

19. Какое из этих устройств можно назвать коммутатором: электрический выключатель;

- автоматическая телефонная станция;
- маршрутизатор;
- мост;

• мультиплексор;
• ни одно из названных.

Какие методы используются при мультиплексировании?

21. Объясните различия между разделением среды передачи и мультиплексированием.

22. Опишите, какие основные задачи нужно решить, чтобы обеспечить информационное взаимодействие любой пары абонентов в коммуникационной сети любого типа.

Как представление общего городского трафика в виде нескольких различных потоков позволяет рационализировать управление городским транспортом?

24. Пусть в сети существует несколько маршрутов между двумя конечными узлами A и B. Перечислите достоинства и недостатки следующих вариантов передачи данных между этими узлами:

• использовать все имеющиеся маршруты для параллельной передачи данных;

• передавать все данные по одному оптимальному по некоторому критерию маршруту;

• использовать несколько маршрутов из набора всех возможных маршрутов и разделять между ними передаваемые данные.

2 Стеки коммуникационных протоколов

2.1 Модель OSI

Из того, что протокол является соглашением, принятым двумя взаимодействующими объектами, в данном случае двумя работающими в сети компьютерами, совсем не следует, что он обязательно является стандартным. Но на практике при реализации сетей стремятся использовать стандартные протоколы. Это могут быть фирменные, национальные или международные стандарты.

В начале 80-х годов ряд международных организаций по стандартизации – ISO, ITU-T и некоторые другие – разработали модель, которая сыграла значительную роль в развитии сетей. Эта модель называется *моделью взаимодействия открытых систем (OpenSystemInterconnection, OSI)* или моделью OSI. Модель OSI определяет различные уровни взаимодействия систем, дает им стандартные имена и указывает, какие функции должен выполнять каждый уровень. Модель OSI была разработана на основании большого опыта, полученного при создании компьютерных сетей, в основном глобальных, в 70-е годы. Полное описание этой модели занимает более 1000 страниц текста.

2.1.1 Семь уровней модели OSI

В модели OSI (рисунок2.1) средства взаимодействия делятся на семь уровней: прикладной, представительный, сеансовый, транспортный, сетевой, канальный и физический. Каждый уровень имеет дело с одним определенным аспектом взаимодействия сетевых устройств.

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

Следует также иметь в виду, что приложение может взять на себя функции некоторых верхних уровней модели OSI. Например,

некоторые СУБД имеют встроенные средства удаленного доступа к файлам. В этом случае приложение, выполняя доступ к удаленным ресурсам, не использует системную файловую службу; оно обходит верхние уровни модели OSI и обращается напрямую к системным средствам, ответственным за транспортировку сообщений по сети, которые располагаются на нижних уровнях модели OSI.

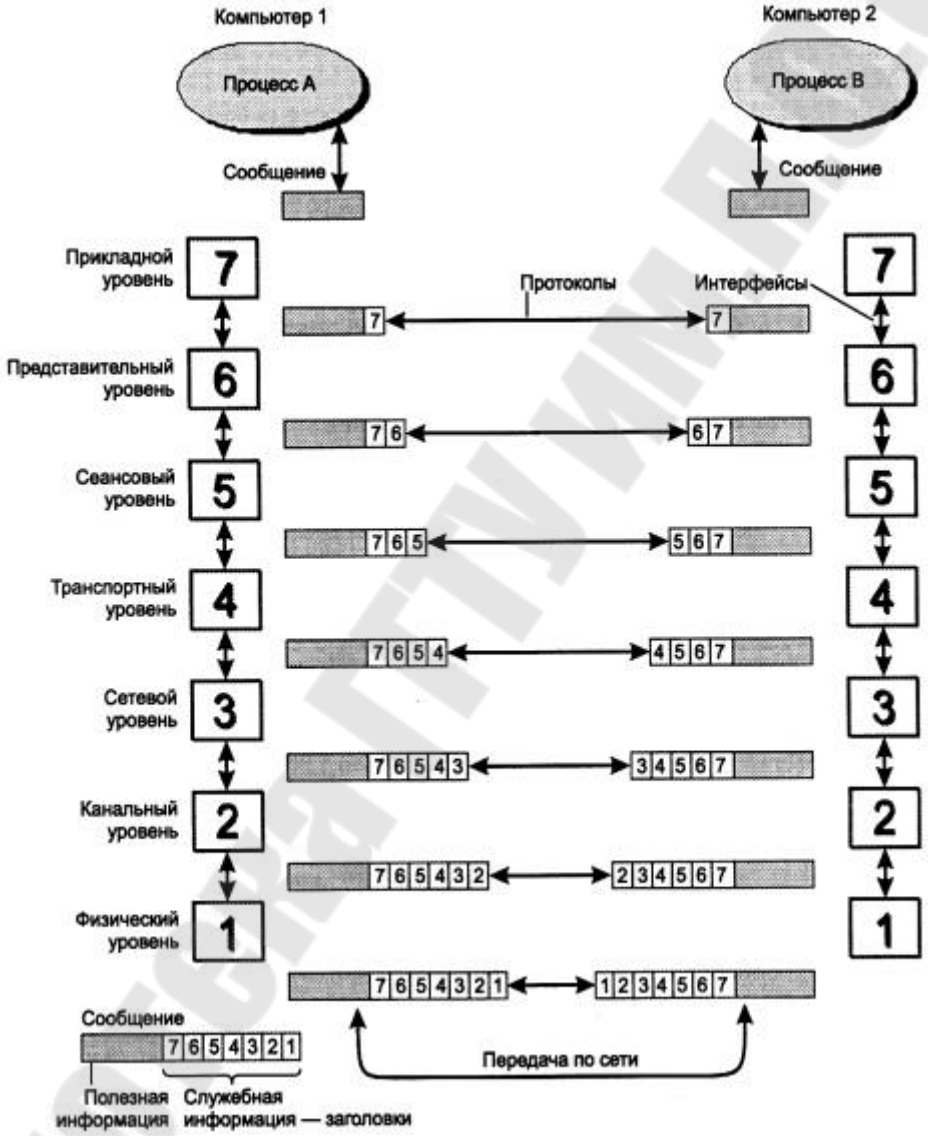


Рисунок 2.1– Модель взаимодействия открытых систем ISO/OSI

Итак, пусть приложение обращается с запросом к прикладному уровню, например к файловой службе. На основании этого запроса программное обеспечение прикладного уровня формирует сообщение стандартного формата. Обычное сообщение состоит из заголовка и поля данных. Заголовок содержит служебную информацию, которую

необходимо передать через сеть прикладному уровню машины-адресата, чтобы сообщить ему, какую работу надо выполнить. В нашем случае заголовок, очевидно, должен содержать информацию о месте нахождения файла и о типе операции, которую необходимо над ним выполнить. Поле данных сообщения может быть пустым или содержать какие-либо данные, например те, которые необходимо записать в удаленный файл. Но для того чтобы доставить эту информацию по назначению, предстоит решить еще много задач, ответственность за которые несут нижележащие уровни.

После формирования сообщения прикладной уровень направляет его вниз по стеку представительному уровню. Протокол представительного уровня на основании информации, полученной из заголовка прикладного уровня, выполняет требуемые действия и добавляет к сообщению собственную служебную информацию – заголовок представительного уровня, в котором содержатся указания для протокола представительного уровня машины-адресата. Полученное в результате сообщение передается вниз сеансовому уровню, который в свою очередь добавляет свой заголовок, и т. д. (Некоторые реализации протоколов помещают служебную информацию не только в начале сообщения в виде заголовка, но и в конце, в виде так называемого «концевика».) Наконец, сообщение достигает нижнего, физического уровня, который собственно и передает его по линиям связи машине-адресату. К этому моменту сообщение «обрастает» заголовками всех уровней (рисунок2.2).

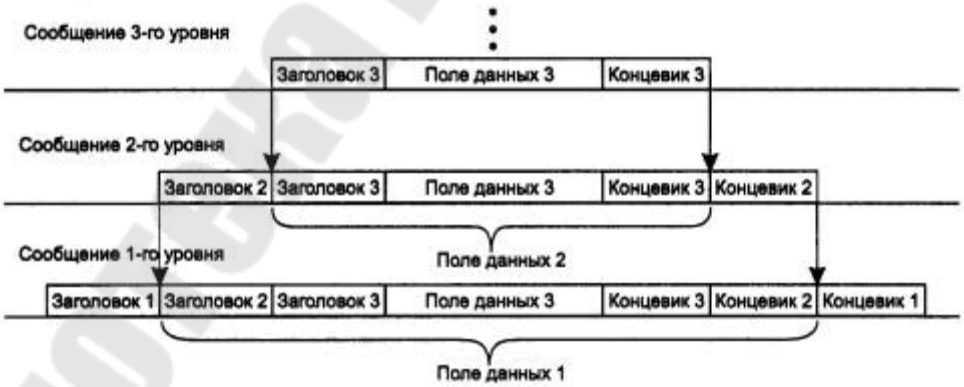


Рисунок 2.2– Вложенность сообщений различных уровней

Когда сообщение по сети поступает на машину – адресат, оно принимается ее физическим уровнем и последовательно перемещается вверх с уровня на уровень. Каждый уровень анализирует и обрабатывает заголовок своего уровня, выполняя

соответствующие данному уровню функции, а затем удаляет этот заголовок и передает сообщение вышележащему уровню.

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

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

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

Остановимся подробнее на каждом из семи уровней модели OSI.

Физический уровень

Физический уровень (Physicallayer) имеет дело с передачей битов по физическим каналам связи, таким, например, как коаксиальный кабель, витая пара, оптоволоконный кабель или цифровой территориальный канал. 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, передающих дискретную информацию, например, крутизна фронтов импульсов, уровни напряжения или тока передаваемого сигнала, тип кодирования, скорость передачи

сигналов. Кроме этого, здесь стандартизуются типы разъемов и назначение каждого контакта.

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

Примером протокола физического уровня может служить спецификация l0-Base-T технологии Ethernet, которая определяет в качестве используемого кабеля неэкранированную витую пару категории 3 с волновым сопротивлением 100 Ом, разъем RJ-45, максимальную длину физического сегмента 100 метров, манчестерский код для представления данных в кабеле, а также некоторые другие характеристики среды и электрических сигналов.

Канальный уровень

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

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, а также структуры, полученные из них с помощью мостов и коммутаторов. Примерами протоколов канального уровня являются протоколы Ethernet, TokenRing, FDDI, l00VG-AnyLAN.

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

В глобальных сетях, которые редко обладают регулярной топологией, канальный уровень часто обеспечивает обмен сообщениями только между двумя соседними компьютерами, соединенными индивидуальной линией связи. Примерами протоколов «точка-точка» (как часто называют такие протоколы) могут служить широко распространенные протоколы PPP и LAP-B. В таких случаях для доставки сообщений между конечными узлами через всю сеть используются средства сетевого уровня. Именно так организованы сети X.25. Иногда в глобальных сетях функции канального уровня в чистом виде выделить трудно, так как в одном и том же протоколе они объединяются с функциями сетевого уровня. Примерами такого подхода могут служить протоколы технологий ATM и framerelay.

В целом канальный уровень представляет собой весьма мощный и законченный набор функций по пересылке сообщений между узлами сети. В некоторых случаях протоколы канального уровня оказываются самодостаточными транспортными средствами и могут допускать работу поверх них непосредственно протоколов прикладного уровня или приложений, без привлечения средств сетевого и транспортного уровней. Например, существует реализация протокола управления сетью SNMP непосредственно поверх Ethernet, хотя стандартно этот протокол работает поверх сетевого протокола IP и транспортного протокола UDP. Естественно, что применение такой реализации будет ограниченным – она не подходит для составных сетей разных технологий, например Ethernet и X.25, и даже для такой

сети, в которой во всех сегментах применяется Ethernet, но между сегментами существуют петлевид-ные связи. А вот в двухсегментной сети Ethernet, объединенной мостом, реализация SNMP над канальным уровнем будет вполне работоспособна.

Тем не менее для обеспечения качественной транспортировки сообщений в сетях любых топологий и технологий функций канального уровня оказывается недостаточно, поэтому в модели OSI решение этой задачи возлагается на два следующих уровня – сетевой и транспортный.

Сетевой уровень

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

Протоколы канального уровня локальных сетей обеспечивают доставку данных между любыми узлами только в сети с соответствующей типовой топологией, например топологией иерархической звезды. Это очень жесткое ограничение, которое не позволяет строить сети с развитой структурой, например, сети, объединяющие несколько сетей предприятия в единую сеть, или высоконадежные сети, в которых существуют избыточные связи между узлами. Можно было бы усложнять протоколы канального уровня для поддержания петлевидных избыточных связей, но принцип разделения обязанностей между уровнями приводит к другому решению. Чтобы с одной стороны сохранить простоту процедур передачи данных для типовых топологий, а с другой допустить использование произвольных топологий, вводится дополнительный сетевой уровень.

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

Внутри сети доставка данных обеспечивается соответствующим канальным уровнем, а вот доставкой данных между сетями

занимается сетевой уровень, который и поддерживает возможность правильного выбора маршрута передачи сообщения даже в том случае, когда структура связей между составляющими сетями имеет характер, отличный от принятого в протоколах канального уровня. Сети соединяются между собой специальными устройствами, называемыми маршрутизаторами. *Маршрутизатор* – это устройство, которое собирает информацию о топологии межсетевых соединений и на ее основании пересылает пакеты сетевого уровня в сеть назначения. Чтобы передать сообщение от отправителя, находящегося в одной сети, получателю, находящемуся в другой сети, нужно совершить некоторое количество *транзитных передач между сетями*, или *хопов* (от *hop* – прыжок), каждый раз выбирая подходящий маршрут. Таким образом, маршрут представляет собой последовательность маршрутизаторов, через которые проходит пакет.

На рисунке2.3 показаны четыре сети, связанные тремя маршрутизаторами. Между узлами A и B данной сети пролегают два маршрута: первый через маршрутизаторы 1 и 3, а второй через маршрутизаторы 1, 2 и 3.

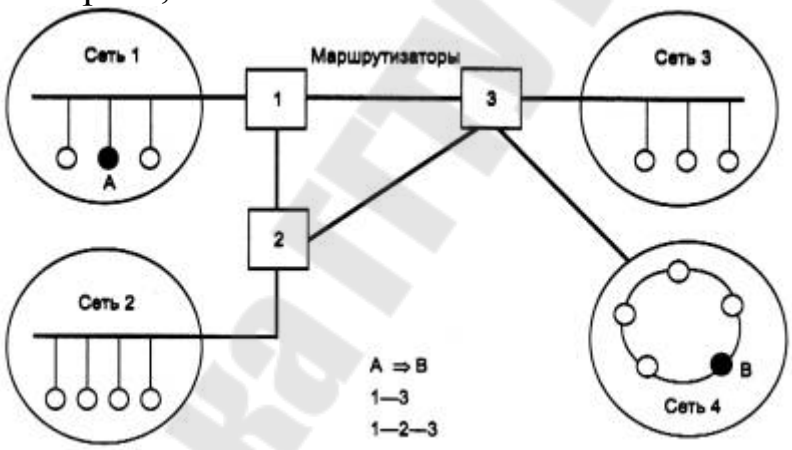


Рисунок 2.3– Пример составной сети

Проблема выбора наилучшего пути называется *маршрутизацией*, и ее решение является одной из главных задач сетевого уровня. Эта проблема осложняется тем, что самый короткий путь не всегда самый лучший. Часто критерием при выборе маршрута является время передачи данных по этому маршруту; оно зависит от пропускной способности каналов связи и интенсивности трафика, которая может изменяться с течением времени. Некоторые алгоритмы маршрутизации пытаются приспособиться к изменению нагрузки, в то время как другие принимают решения на основе средних

показателей за длительное время. Выбор маршрута может осуществляться и по другим критериям, например надежности передачи.

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

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

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

На сетевом уровне работают протоколы еще одного типа, которые отвечают за отображение адреса узла, используемого на сетевом уровне, в локальный адрес сети. Такие протоколы часто называют *протоколами разрешения адресов – AddressResolutionProtocol, ARP*. Иногда их относят не к сетевому уровню, а к канальному, хотя тонкости классификации не изменяют их сути. Примерами протоколов сетевого уровня являются протокол межсетевого взаимодействия IP стека TCP/IP и протокол межсетевого обмена пакетами IPX стека Novell.

Транспортный уровень

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

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

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

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

Сеансовый уровень

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

Представительный уровень

Представительный уровень (PresentationLayer) имеет дело с формой представления передаваемой по сети информации, не меняя при этом ее содержания. За счет уровня представления информация, передаваемая прикладным уровнем одной системы, всегда понятна прикладному уровню другой системы. С помощью средств данного уровня протоколы прикладных уровней могут преодолеть синтаксические различия в представлении данных или же различия в кодах символов, например кодов ASCII и EBCDIC. На этом уровне может выполняться шифрование и дешифрование данных, благодаря которому секретность обмена данными обеспечивается сразу для всех прикладных служб. Примером такого протокола является протокол SecureSocketLayer (SSL), который обеспечивает секретный обмен сообщениями для протоколов прикладного уровня стека TCP/IP.

Прикладной уровень

Прикладной уровень (ApplicationLayer) – 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 служб прикладного уровня. Приведем в качестве примера хотя бы несколько наиболее распространенных реализации файловых служб: NCP в операционной системе NovellNetWare, SMB в MicrosoftWindows NT, NFS, FTP и TFTP, входящие в стек TCP/IP.

Сетезависимые и сетенезависимые уровни

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

Три нижних уровня – физический, канальный и сетевой – являются сетезависимыми, то есть протоколы этих уровней тесно связаны с технической реализацией сети и используемым коммуникационным оборудованием. Например, переход на оборудование FDDI означает полную смену протоколов физического и канального уровней во всех узлах сети.

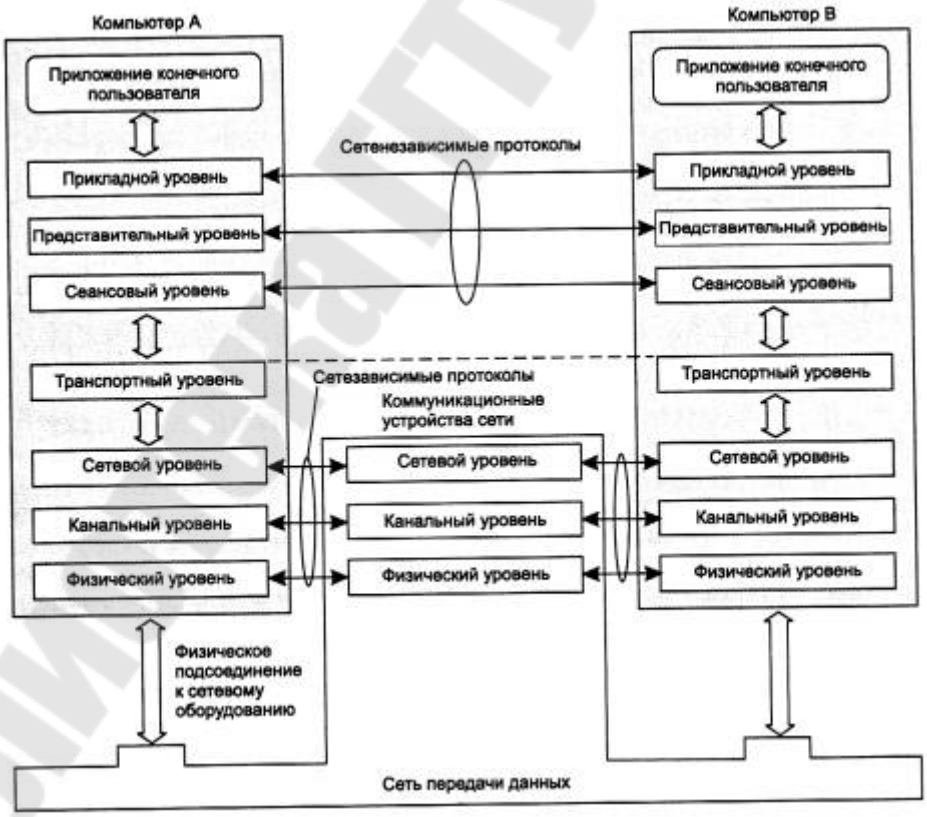


Рисунок 2.4 – Сетезависимые и сетенезависимые уровни модели OSI

Три верхних уровня – прикладной, представительный и сеансовый – ориентированы на приложения и мало зависят от технических особенностей построения сети. На протоколы этих уровней не влияют какие бы то ни было изменения в топологии сети, замена оборудования или переход на другую сетевую технологию. Так, переход от Ethernet на высокоскоростную технологию l00VG-AnyLAN не потребует никаких изменений в программных средствах, реализующих функции прикладного, представительного и сеансового уровней.

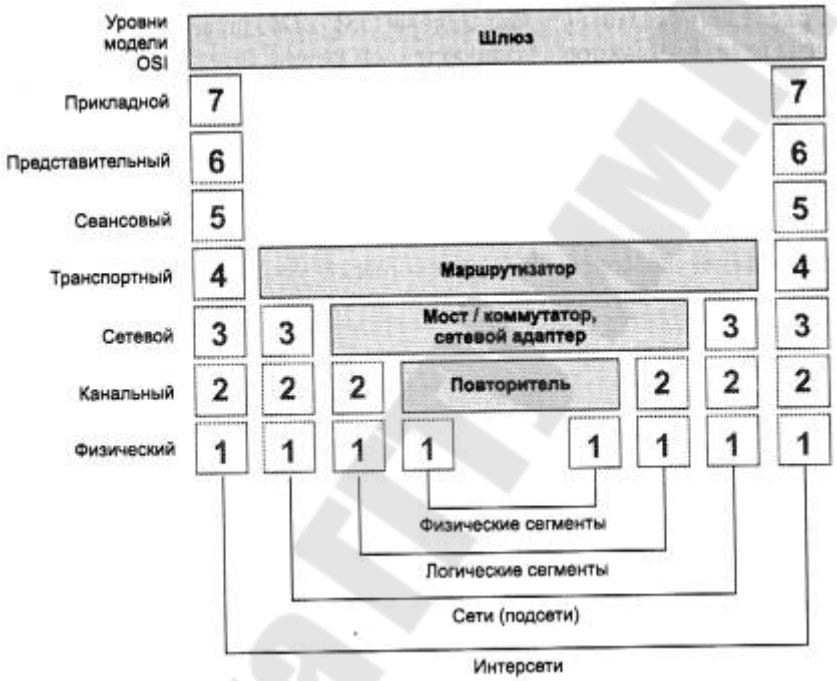


Рисунок 2.5– Соответствие функций различных устройств сети уровням модели OSI.

Транспортный уровень является промежуточным, он скрывает все детали функционирования нижних уровней от верхних. Это позволяет разрабатывать приложения, не зависящие от технических средств непосредственной транспортировки сообщений. На рисунке 2.4 показаны уровни модели OSI, на которых работают различные элементы сети. Компьютер с установленной на нем сетевой ОС взаимодействует с другим компьютером с помощью протоколов всех семи уровней. Это взаимодействие компьютеры осуществляют опосредовано через различные коммуникационные устройства: концентраторы, модемы, мосты, коммутаторы, маршрутизаторы, мультиплексоры. В зависимости от типа коммуникационное

устройство может работать либо только на физическом уровне (повторитель), либо на физическом и канальном (мост), либо на физическом, канальном и сетевом, иногда захватывая и транспортный уровень (маршрутизатор). На рисунке 2.5 показано соответствие функций различных коммуникационных устройств уровням модели OSI.

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, службами, поддерживаемыми на верхних уровнях, и прочими параметрами.

2.1.2 Понятие «открытая система»

Модель OSI, как это следует из ее названия (OpenSystemInterconnection), описывает взаимосвязи открытых систем. Что же такое открытая система?

В широком смысле *открытой системой* может быть названа любая система (компьютер, вычислительная сеть, ОС, программный пакет, другие аппаратные и программные продукты), которая построена в соответствии с открытыми спецификациями.

Напомним, что под термином «спецификация» (в вычислительной технике) понимают формализованное описание аппаратных или программных компонентов, способов их функционирования, взаимодействия с другими компонентами, условий эксплуатации, ограничений и особых характеристик. Понятно, что не всякая спецификация является стандартом. В свою очередь, под открытыми спецификациями понимаются опубликованные, общедоступные спецификации, соответствующие стандартам и принятые в результате достижения согласия после всестороннего обсуждения всеми заинтересованными сторонами.

Использование при разработке систем открытых спецификаций позволяет третьим сторонам разрабатывать для этих систем различные аппаратные или программные средства расширения и модификации, а также создавать программно-аппаратные комплексы из продуктов разных производителей.

Для реальных систем полная открытость является недостижимым идеалом. Как правило, даже в системах, называемых открытыми, этому определению соответствуют лишь некоторые части, поддерживающие внешние интерфейсы. Например, открытость семейства операционных систем Unix заключается, кроме всего

прочего, в наличии стандартизованного программного интерфейса между ядром и приложениями, что позволяет легко переносить приложения из среды одной версии Unix в среду другой версии. Еще одним примером частичной открытости является применение в достаточно закрытой операционной системе NovellNetWare открытого интерфейса OpenDriverInterface (ODI) для включения в систему драйверов сетевых адаптеров независимых производителей. Чем больше открытых спецификаций использовано при разработке системы, тем более открытой она является.

Модель OSI касается только одного аспекта открытости, а именно открытости средств взаимодействия устройств, связанных в компьютерную сеть. Здесь под открытой системой понимается сетевое устройство, готовое взаимодействовать с другими сетевыми устройствами с использованием стандартных правил, определяющих формат, содержание и значение принимаемых и отправляемых сообщений.

Если две сети построены с соблюдением принципов открытости, то это дает следующие преимущества:

- возможность построения сети из аппаратных и программных средств различных производителей, придерживающихся одного и того же стандарта;
- возможность безболезненной замены отдельных компонентов сети другими, более совершенными, что позволяет сети развиваться с минимальными затратами;
- возможность легкого сопряжения одной сети с другой;
- простота освоения и обслуживания сети.

Ярким примером открытой системы является международная сеть Internet. Эта сеть развивалась в полном соответствии с требованиями, предъявляемыми к открытым системам. В разработке ее стандартов принимали участие тысячи специалистов-пользователей этой сети из различных университетов, научных организаций и фирм-производителей вычислительной аппаратуры и программного обеспечения, работающих в разных странах. Само название стандартов, определяющих работу сети Internet – RequestForComments (RFC), что можно перевести как «запрос на комментарии», – показывает гласный и открытый характер принимаемых стандартов. В результате сеть Internet сумела объединить в себе самое разнообразное оборудование и программное обеспечение огромного числа сетей, разбросанных по всему миру.

2.1.3 Модульность и стандартизация

Модульность – это одно из неотъемлемых и естественных свойств вычислительных сетей. Модульность проявляется не только в многоуровневом представлении коммуникационных протоколов в конечных узлах сети, хотя это, безусловно, важная и принципиальная особенность сетевой архитектуры. Сеть состоит из огромного числа различных модулей – компьютеров, сетевых адаптеров, мостов, маршрутизаторов, модемов, операционных систем и модулей приложений. Разнообразные требования, предъявляемые предприятиями к компьютерным сетям, привели к такому же разнообразию выпускаемых для построения сети устройств и программ. Эти продукты отличаются не только основными функциями (имеются в виду функции, выполняемые, например, повторителями, мостами или программными редиректорами), но и многочисленными вспомогательными функциями, предоставляющими пользователям или администраторам дополнительные удобства, такие как автоматизированное конфигурирование параметров устройства, автоматическое обнаружение и устранение некоторых неисправностей, возможность программного изменения связей в сети и т. п. Разнообразие увеличивается также потому, что многие устройства и программы отличаются сочетаниями тех или иных основных и дополнительных функций – существуют, например, устройства, сочетающие основные возможности коммутаторов и маршрутизаторов, к которым добавляется еще и набор некоторых дополнительных функций, характерный только для данного продукта.

В результате не существует компании, которая смогла бы обеспечить производство полного набора всех типов и подтипов оборудования и программного обеспечения, требуемого для построения сети. Но, так как все компоненты сети должны работать согласованно, совершенно необходимым оказалось принятие многочисленных стандартов, которые, если не во всех, то хотя бы в большинстве случаев, гарантировали бы совместимость оборудования и программ различных фирм-изготовителей. Таким образом, понятия модульности и стандартизации в сетях неразрывно связаны, и модульный подход только тогда дает преимущества, когда он сопровождается следованием стандартам.

В результате открытый характер стандартов и спецификаций важен не только для коммуникационных протоколов, но и для всех

многочисленных функций разнообразных устройств и программ, выпускаемых для построения сети. Нужно отметить, что большинство стандартов, принимаемых сегодня, носят открытый характер. Время закрытых систем, точные спецификации на которые были известны только фирме-производителю, ушло. Все осознали, что возможность легкого взаимодействия с продуктами конкурентов не снижает, а наоборот, повышает ценность изделия, так как его можно применить в большем количестве работающих сетей, построенных на продуктах разных производителей. Поэтому даже фирмы, ранее выпускавшие весьма закрытые системы – такие как IBM, Novell или Microsoft, – сегодня активно участвуют в разработке открытых стандартов и применяют их в своих продуктах.

Сегодня в секторе сетевого оборудования и программ с совместимостью продуктов разных производителей сложилась следующая ситуация. Практически все продукты, как программные, так и аппаратные, совместимы по функциям и свойствам, которые были внедрены в практику уже достаточно давно и стандарты на которые уже разработаны и приняты по крайней мере 3-4 года назад. В то же время очень часто принципиально новые устройства, протоколы и свойства оказываются несовместимыми даже у ведущих производителей. Такая ситуация наблюдается не только для тех устройств или функций, стандарты на которые еще не успели принять (это естественно), но и для устройств, стандарты на которые существуют уже несколько лет. Совместимость достигается только после того, как все производители реализуют этот стандарт в своих изделиях, причем одинаковым образом.

2.1.4 Источники стандартов

Работы по стандартизации вычислительных сетей ведутся большим количеством организаций. В зависимости от статуса организаций различают следующие виды стандартов:

- *стандарты отдельных фирм* (например, стек протоколов DECnet фирмы DigitalEquipment или графический интерфейс OPEN LOOK для Unix-систем фирмы Sun);
- *стандарты специальных комитетов и объединений*, создаваемых несколькими фирмами, например стандарты технологии ATM, разрабатываемые специально созданным объединением ATM Forum, насчитывающем около 100 коллективных участников, или стандарты союза FastEthernetAlliance по разработке стандартов 100 Мбит Ethernet;

- *национальные стандарты*, например, стандарт FDDI, представляющий один из многочисленных стандартов, разработанных Американским национальным институтом стандартов (ANSI), или стандарты безопасности для операционных систем, разработанные Национальным центром компьютерной безопасности (NCSC) Министерства обороны США;
- *международные стандарты*, например, модель и стек коммуникационных протоколов Международной организации по стандартам (ISO), многочисленные стандарты Международного союза электросвязи (ITU), в том числе стандарты на сети с коммутацией пакетов X.25, сети framerelay, ISDN, модемы и многие другие.

Некоторые стандарты, непрерывно развиваясь, могут переходить из одной категории в другую. В частности, фирменные стандарты на продукцию, получившую широкое распространение, обычно становятся международными стандартами де-факто, так как вынуждают производителей из разных стран следовать фирменным стандартам, чтобы обеспечить совместимость своих изделий с этими популярными продуктами. Например, из-за феноменального успеха персонального компьютера компании IBM фирменный стандарт на архитектуру IBM PC стал международным стандартом де-факто.

Более того, ввиду широкого распространения некоторые фирменные стандарты становятся основой для национальных и международных стандартов де-юре. Например, стандарт Ethernet, первоначально разработанный компаниями DigitalEquipment, Intel и Xerox, через некоторое время и в несколько измененном виде был принят как национальный стандарт IEEE 802.3, а затем организация ISO утвердила его в качестве международного стандарта ISO 8802.3.

2.2 Стандартные стеки коммуникационных протоколов

Важнейшим направлением стандартизации в области компьютерных сетей является стандартизация коммуникационных протоколов. В настоящее время в сетях используется большое количество стеков коммуникационных протоколов. Наиболее популярными являются стеки: TCP/IP, IPX/SPX, NetBIOS/SMB, DECnet, SNA и OSI. Все эти стеки, кроме SNA на нижних уровнях – физическом и канальном, – используют одни и те же хорошо стандартизованные протоколы Ethernet, TokenRing, FDDI и некоторые другие, которые позволяют использовать во всех сетях одну и ту же

аппаратуру. Зато на верхних уровнях все стеки работают по своим собственным протоколам. Эти протоколы часто не соответствуют рекомендуемому моделью OSI разбиению на уровни. В частности, функции сеансового и представительного уровня, как правило, объединены с прикладным уровнем. Такое несоответствие связано с тем, что модель OSI появилась как результат обобщения уже существующих и реально используемых стеков, а не наоборот.

Стек TCP/IP

Стек TCP/IP был разработан по инициативе Министерства обороны США более 20 лет назад для связи экспериментальной сети ARPAnet с другими сетями как набор общих протоколов для разнородной вычислительной среды. Большой вклад в развитие стека TCP/IP, который получил свое название по популярным протоколам IP и TCP, внес университет Беркли, реализовав протоколы стека в своей версии ОС UNIX. Популярность этой операционной системы привела к широкому распространению протоколов TCP, IP и других протоколов стека. Сегодня этот стек используется для связи компьютеров всемирной информационной сети Internet, а также в огромном числе корпоративных сетей.

Стек TCP/IP на нижнем уровне поддерживает все популярные стандарты физического и канального уровней: для локальных сетей – это Ethernet, TokenRing, FDDI, для глобальных – протоколы работы на аналоговых коммутируемых и выделенных линиях SLIP, PPP, протоколы территориальных сетей X.25 и ISDN.

Основными протоколами стека, давшими ему название, являются протоколы IP и TCP. Эти протоколы в терминологии модели OSI относятся к сетевому и транспортному уровням соответственно. IP обеспечивает продвижение пакета по составной сети, а TCP гарантирует надежность его доставки.

За долгие годы использования в сетях различных стран и организаций стек TCP/IP вобрал в себя большое количество протоколов прикладного уровня. К ним относятся такие популярные протоколы, как протокол пересылки файлов FTP, протокол эмуляции терминала telnet, почтовый протокол SMTP, используемый в электронной почте сети Internet, гипертекстовые сервисы службы WWW и многие другие.

Сегодня стек TCP/IP является самым распространенным стеком транспортных протоколов компьютерных сетей. Действительно, в сети Internet объединено не менее миллиарда компьютеров по всему

миру, которые взаимодействуют друг с другом с помощью стека протоколов TCP/IP.

Стремительный рост популярности Internet привел и к изменениям в расстановке сил в мире коммуникационных протоколов – протоколы TCP/IP, на которых построен Internet, стали быстро теснить бесспорного лидера прошлых лет – стек IPX/SPX компании Novell.

Хотя протоколы TCP/IP неразрывно связаны с Internet и каждый из всей армады компьютеров Internet работает на основе этого стека, существует большое количество локальных, корпоративных и территориальных сетей, непосредственно не являющихся частями Internet, в которых также используют протоколы TCPДР. Чтобы отличать их от Internet, эти сети называют сетями TCP/IP или просто IP-сетями.

Поскольку стек TCP/IP изначально создавался для глобальной сети Internet, он имеет много особенностей, дающих ему преимущество перед другими протоколами, когда речь заходит о построении сетей, включающих глобальные связи. В частности, очень полезным свойством, делающим возможным применение этого протокола в больших сетях, является его способность фрагментировать пакеты. Действительно, большая составная сеть часто состоит из сетей, построенных на совершенно разных принципах. В каждой из этих сетей может быть установлена собственная величина максимальной длины единицы передаваемых данных (кадра). В таком случае при переходе из одной сети, имеющей большую максимальную длину, в сеть с меньшей максимальной длиной может возникнуть необходимость деления передаваемого кадра на несколько частей. Протокол IP стека TCP/IP эффективно решает эту задачу.

Другой особенностью технологии TCP/IP является гибкая система адресации, позволяющая более просто по сравнению с другими протоколами аналогичного назначения включать в интерсеть сети других технологий. Это свойство также способствует применению стека TCP/IP для построения больших гетерогенных сетей.

В стеке TCP/IP очень экономно используются возможности широковещательных рассылок. Это свойство совершенно необходимо при работе на медленных каналах связи, характерных для территориальных сетей.

Однако, как и всегда, за получаемые преимущества надо платить, и платой здесь оказываются высокие требования к ресурсам и сложность администрирования IP-сетей. Мощные функциональные возможности протоколов стека TCP/IP требуют для своей реализации высоких вычислительных затрат. Гибкая система адресации и отказ от широковещательных рассылок приводят к наличию в IP-сети различных централизованных служб типа DNS, DHCP и т. п. Каждая из этих служб направлена на облегчение администрирования сети, в том числе и на облегчение конфигурирования оборудования, но в то же время сама требует пристального внимания со стороны администраторов.

Можно приводить и другие доводы за и против стека протоколов Internet, однако факт остается фактом – сегодня это самый популярный стек протоколов, широко используемый как в глобальных, так и локальных сетях.

Вопросы для самоконтроля.

1. Что такое «открытая система»? Приведите примеры закрытых систем. Чем отличаются открытые и закрытые системы на примере социальных организаций.
2. Поясните разницу в употреблении терминов «протокол» и «интерфейс» применительно к многоуровневой модели взаимодействия устройств в сети.
3. Что стандартизует модель OSI?
4. Что стандартизует стек OSI?
5. Почему в модели OSI семь уровней?
6. Дайте краткое описание функций каждого уровня и приведите примеры стандартных протоколов для каждого уровня модели OSI.
7. Какие аналоги основных принципов от крытой модели OSI можно обнаружить в процессе общения двух человек?
8. Являются ли термины «спецификация» и «стандарт» синонимами?
9. Благодаря чему глобальная сеть Интернет смогла достичь таких грандиозных масштабов распространения? Какова роль стека TCP/IP?

3 Глобальная компьютерная сеть Интернет

3.1 Основные определения

24 октября 1995 года Федеральный сетевой совет (FNC), США, единодушно одобрил резолюцию, определяющую термин "Интернет" Это определение разрабатывалось при участии специалистов в области сетей и в области прав на интеллектуальную собственность.

Интернет — это глобальная информационная система, которая:

1.логически взаимосвязана пространством глобальных уникальных адресов, основанных на Интернет-протоколе (IP) или на последующих расширениях или преемниках IP;

2.способна поддерживать коммуникации с использованием семейства Протокола управления передачей, который называется Интернет-протоколом (TCP/IP) или его последующих расширений/преемников и/или других IP-совместимых протоколов;

3.обеспечивает, использует или делает доступной, на общественной или частной основе, высокоуровневые сервисы, надстроенные над описанной здесь коммуникационной и иной связанной с ней инфраструктурой.

Как видно из определения, в основе сети Интернет лежит использование протокола сетевого уровня, IP- протокола, над которым должны работать протоколы более высокого уровня, в первую очередь TCP – протокол.

Следует отметить, что революционизирующее влияние Интернет на мир компьютеров и коммуникаций не имеет исторических аналогов. Изобретение телеграфа, телефона, радио и компьютера подготовило почву для происходящей ныне их беспрецедентной интеграции. Интернет одновременно является и средством общемирового вещания, и механизмом распространения информации, и средой для сотрудничества и общения людей и компьютеров, охватывающей весь земной шар.

Интернет представляет собой один из наиболее успешных примеров того, какую пользу могут принести долгосрочные вложения и поддержка исследований и разработки информационной инфраструктуры. Начиная с ранних исследований в области пакетной коммутации, правительства различных стран, промышленность и академическая наука оставались партнерами в развитии и развертывании этой новой сетевой технологии.

В историческом развитии сети Интернет можно выделить четыре различных аспекта:

- технологическая эволюция исследований по пакетной коммутации;
- развитие методов и средств эксплуатации и управления глобальной и сложной сетевой инфраструктурой;
- социальный аспект, приведший к образованию широкого сообщества пользователей;
- коммерциализация, характеризуемая чрезвычайно эффективным превращением результатов исследований в развернутую, широко доступную информационную систему.

3.2 Зарождение Интернет

У истоков создания сети Интернет стояла группа ученых и инженеров Управления перспективных исследований и разработок Министерства обороны США – DARPA (DefenseAdvancedResearchAgency), созданная в 1962 году под руководством Дж. Ликлайдера. Этим ученым впервые была сформулирована концепция «галактической сети», объединяющая огромное количество компьютеров, и с помощью которой каждый пользователь сможет быстро получить доступ к данным и программам, расположенным на любом компьютере. Эта концепция очень близка по духу современному состоянию Интернет. Одновременно появились работы Леонарда Клейнрока по теории коммутации пакетов (пакетной коммутации), в которых теоретически обосновывалось возможность создания компьютерных сетей на основе пакетной коммутации. В дальнейшем проведенные эксперименты показали, что компьютеры с разделением времени могут успешно работать вместе, выполняя программы и осуществляя выборку на удаленной машине. Стало ясно и то, что телефонная система того времени с коммутацией соединений абсолютно непригодна для создания компьютерной сети.

В 1967 году появился проект первой компьютерной сети ARPANET, а в 1968 были доработана структура и спецификации этой сети, которая должна была работать по технологии коммутации пакетов. После разработки первого коммутатора пакетов компанией BBN, который назывался тогда интерфейсным процессором, появилась возможность провести соединения с их помощью

нескольких компьютеров, находящихся на большом расстоянии друг от друга. В сентябре 1969 один из коммутаторов был установлен в Калифорнийском университете, к нему был подключен компьютер, а второй коммутатор с подключенным компьютером разместили в Стэндфордском исследовательском институте. Через месяц было послано первое компьютерное сообщение из Калифорнийского университета, которое было успешно принято в Стэнфорде. Двумя следующими узлами ARPANET стали университет города Санта-Барбара и Университет штат Юта.

Таким образом, к концу 1969 года первые четыре компьютера были объединены в первоначальную конфигурацию ARPANET. В последующие годы число компьютеров, подключенных к ARPANET, быстро росло.

Одновременно велись работы по созданию функционально полного протокола межкомпьютерного взаимодействия и другого сетевого программного обеспечения. В декабре 1970 года Сетевая рабочая группа (NetworkWorkingGroup, NWG) завершила работу над первой версией протокола, получившего название Протокол управления сетью (NetworkControlProtocol, NCP). После того, как в 1971- 1972 годах этот протокол был реализован на всех узлах ARPANET, пользователи сети смогли приступить к разработке приложений работающих над этим протоколом.

В марте 1972 года появилось *первое такое приложение – электронная почта*. Создателем программы электронной почты стал сотрудник вышеупомянутой компании BBNРэйТомлисон (RayTomlinson), он же предложил использовать значок @ («собака»). Для своего времени электронная почта стала тем, же чем в наши дни является служба WWW- исключительно мощным катализатором роста всех видов межперсональных потоков данных.

3.3 Концепция объединения сетей

Интернет основывается на идее существования множества независимых сетей почти произвольной архитектуры, начиная от ARPANET. Интернет в современном понимании воплощает ключевой технический принцип *открытости сетевой архитектуры*. При подобном подходе архитектура и техническая реализация отдельных сетей не навязываются извне – они могут свободно выбираться

поставщиком сетевых услуг при сохранении возможности объединения с другими сетями посредством сетевого уровня.

Открытая сетевая архитектура подразумевает, что отдельные сети могут проектироваться и разрабатываться независимо, со своими уникальными интерфейсами, предоставляемыми пользователям и/или другим поставщикам сетевых услуг, включая услуги Интернет. При проектировании каждой сети могут быть приняты во внимание специфика окружения и особые требования пользователей. Вообще говоря, не накладывается никаких ограничений на типы объединяемых сетей или их территориальный масштаб.

Как уже указывалось выше, в сети ARPANET использовался протокол NCP.

Однако NCP не содержал средств для адресации сетей и отдельных машин. В обеспечении сквозной надежности протокол NCP полагался на хорошие линии связи.. Если какие-то пакеты терялись, протокол и поддерживаемые им приложения должны были остановиться. В модели NCP отсутствовало сквозное управление ошибками, поскольку ARPANET должна была являться единственной существующей сетью, причем настолько надежной, что от компьютеров не требовалось умение реагировать на ошибки. Таким образом, протокол NCP не соответствовал требованиям открытой сетевой архитектуры и требовал серьезной доработки.

СотрудникDARPA Роберт Канн в 1972 году предложил разработать новую версию протокола, удовлетворяющую требованиям окружения с открытой сетевой архитектурой.

Этот протокол позднее будет назван ***TransmissionControlProtocol/ InternetProtocol (TCP/IP — Протокол управления передачей/Межсетевой протокол)***.

В то время как NCP действовал как драйвер устройства, новинка должна была в большей мере напоминать коммуникационный протокол.

В основе разработки нового протокола лежали четыре принципа:

- Каждая сеть должна сохранять свою индивидуальность. При подключении кИнтернет сети не должны подвергаться внутренним переделкам;

• Передача пакетов должна идти по принципу "максимум возможного". Если пакет не прибыл в пункт назначения, источник должен вскоре повторно передать его;

• Для связывания сетей должны использоваться черные ящики; позднее их назовут *шлюзами и маршрутизаторами;*

• На локальном уровне не должно существовать глобальной системы управления.

Самыми первыми результатами по реализации указанных принципов стало:

• Общение между двумя компьютерами логически должно представляться как обмен непрерывными последовательностями байт. Для идентификации байта используется его позиция в последовательности.

• Управление потоком данных осуществляется на основе механизмов подтверждений. Получатель может выбирать, когда посылать подтверждение, распространяющееся на все полученные к этому моменту пакеты.

В публикациях того времени по объединению сетей (начало 70-х годов) первоначально описывался один протокол, названный TCP. Он предоставлял все услуги по транспортировке и перенаправлению данных в Интернет. Планировалось, что протокол TCP будет поддерживать целый диапазон транспортных сервисов. Затем, однако, протокол TCP был раздел на два протокола — простой IP, обслуживающий только адресацию и перенаправление отдельных пакетов, и отдельный TCP, имеющий дело с такими аспектами, как управление потоком данных и нейтрализация потери пакетов. Для приложений, не нуждавшихся в услугах TCP, была добавлена альтернатива — Пользовательский дэйтаграммный протокол (User DatagramProtocol, UDP), открывающий прямой доступ к базовым сервисам уровня IP с приложений верхнего уровня.

Ключевая концепция создания Интернет состояла в том, что объединение сетей проектировалось не для какого-то одного приложения, но как универсальная инфраструктура, над которой могут быть надстроены новые приложения. Основой этих приложений являлся протокол TCP / IP.

Широкое распространение в 1980-е годы локальных сетей, персональных компьютеров и рабочих станций дало толчок бурному росту Интернет. Технология Ethernet, разработанная в 1973 году фирмой Xerox PARC, в наши дни является доминирующей сетевой

технологией в Интернет, а персональные компьютеры и рабочие станции стали доминирующими компьютерами.

Рост Интернет вызвал важные изменения и в вопросах управления. Чтобы сделать сеть более дружественной для человека, компьютерам были присвоены имена, делающие ненужным запоминание числовых адресов. Пол Мокапетрис (PaulMockapetris) из Института информатики Университета Южной Калифорнии придумал доменную систему имен (DomainNameSystem, DNS). DNS позволила создать масштабируемый распределенный механизм для отображения иерархических имен компьютеров (например, www.acm.org) в Интернет-адреса.

Еще одной особенностью, вызванной ростом Интернет, стало внесение изменений в программное обеспечение. Протокол TCP/IP стал встраиваться в существующие операционные системы Unix.

В целом стратегия встраивания протоколов Интернет TCP/IPв самую распространенную операционную систему, явилась одним из ключевых элементов успешного и повсеместного распространения Интернет.

Протокол TCP/IP был принят в качестве военного стандарта в 1980 году. Это позволило военным начать использование технологической базы Интернет и, в конце концов, привело к разделению на военное и гражданское Интернет-сообщества. К 1983 году ARPANET использовало значительное число военных исследовательских, разрабатывающих и эксплуатирующих организаций.

Кроме этого к 1985 году технологии Интернет поддерживались широкими кругами исследователей и разработчиков. Интернет начинали использовать для повседневных компьютерных коммуникаций люди самых разных категорий. Особую популярность завоевала электронная почта, работавшая на разных платформах. Совместимость различных почтовых систем продемонстрировала выгоды массовых электронных коммуникаций между людьми.

Громадным шагом в развитии Интернет стала разработка в 1989 году Тимом БернерсомЛи гипертекстовой среды, а также разработка им первого Web-браузера, который назывался WorldWideWeb.

17 мая 1991 года на вычислительных системах Европейской физической лаборатории CERN (EuropeanOrganizationforNuclearresearch) была установлена окончательная версия первого в мире Web-сервера.

3.4 Создание инфраструктуры Интернет

К середине 1970-х годов компьютерные сети начали расти, как грибы после дождя. Министерство энергетики США сначала создало сеть MFENet в интересах исследователей термоядерного синтеза с магнитным удержанием, затем специалисты в области физики высоких энергий получили сеть HEPNet, для астрофизиков из NASA построили сеть SPAN, национальный научный фонд (NSF), США, развернул сеть CSNET, объединившую специалистов по информатике из академических и промышленных кругов. Указанные сети должны были использоваться замкнутым сообществом специалистов; как правило, этим работа сетей и ограничивалась. Особой потребности в совместимости сетей не было; соответственно, не было и самой совместимости. Важным шагом по объединению сетей стало в 1985 году важное решение об обязательном использовании в NSFNet протокола TCP/IP.

Размах сети NSFNet, воспринимаемой уже как сеть Интернет Размеры ее финансирования составили 200 миллионов долларов за период с 1986 по 1995 год. В сочетании с качеством TCP/IP протоколов это привело к тому, что в начале 90-х семейство TCP/IP вытеснило или значительно потеснило во всем мире большинство других протоколов глобальных компьютерных сетей. К 1990 году окончательно разукомплектовали сеть ARPANET, которая не могла уже конкурировать с новыми технологиями Интернет. Протокол IP уверенно становился доминирующим сервисом транспортировки данных в глобальной информационной инфраструктуре.

3.5 Роль документации

Ключом к быстрому росту Интернет стал свободный, открытый доступ к основным документам, особенно к спецификациям протоколов.

ARPANET и Интернет, зародившиеся в университетском исследовательском сообществе, развивались в академических традициях открытой публикации идей и результатов. Однако обычный академический цикл был слишком формальным и

медленным для динамичного обмена идеями, необходимого при создании сетей.

Ключевым шагом для обмена идеями стало опубликование в 1969 году серии публикаций "Запросы на комментарии и предложения" (RequestForComments, RFC). Эти статьи должны были служить целям неформального, быстрого распространения идей и их обсуждения с другими сетевыми специалистами. Первоначально RFC-статьи печатались на бумаге и рассылались обычной медленной почтой. После того, как начал использоваться Протокол передачи файлов (FileTransferProtocol, FTP), RFC-статьи стали готовить в виде файлов и передавать посредством FTP. Сейчас эти документы легко доступны по Всемирной паутине (WorldWideWeb), они лежат на десятках серверов во всех частях света.

RFC-статьи позволили создать положительную обратную связь, когда идеи и предложения, содержавшиеся в одном документе, служили отправной точкой для создания новых документов с новыми идеями, и так далее. Когда достигался определенный уровень согласия (или, по крайней мере, вырабатывался согласованный набор идей), готовились спецификации, служившие основой для реализаций, выполнявшихся несколькими командами исследователей.

Со временем RFC-статьи стали посвящаться в основном стандартам протоколов ("официальным" спецификациям), хотя осталась и определенная доля информационных заметок, описывающих альтернативные подходы или идейные основы протокольных и технических решений. Сейчас RFC-статьи рассматриваются как протокол деятельности по стандартизации и реализации Интернет.

Открытый доступ к документам RFC (бесплатный для всех подключенных кИнтернет) способствовал росту Интернет, поскольку он позволял использовать действующие спецификации и во время занятий со студентами, и в процессе разработки новых систем.

Электронная почта сыграла очень важную роль во всех аспектах жизни Интернет, особенно при разработке спецификаций протоколов, технических стандартов и реализационных решений. Самые первые RFC-статьи зачастую представляли собой набор идей, предлагавшихся на всеобщее обсуждение группой исследователей из какой-то одной местности. Использование электронной почты изменило характер авторства — RFC-статьи стали представляться

коллективами авторов с общими взглядами, не зависящими от территориальной принадлежности.

Для выработки спецификаций протоколов в течение долгого времени использовались списки электронной почтовой рассылки; и поныне они остаются важным рабочим инструментом. Сейчас в иерархии списков насчитывается более 70-ти тематических групп, занимающихся разными аспектами Интернет. Каждая из этих групп имеет список рассылки для обсуждения проектов разрабатываемых документов. После согласования проекта в рабочей группе, он публикуется в виде RFC-документа.

Быстрый нынешний рост Интернет во многом объясняется осознанием выгод от разделения информации, которое обеспечивает сеть. При этом важно понимать, что первым видом информации, разделявшейся в сети, были RFC-документы, описывавшие проектирование и эксплуатацию Интернет. Этот уникальный метод разработки новых сетевых средств остается критически важным для дальнейшей эволюции Интернет.

3.6 Коммерциализация технологии

Коммерциализация Интернет включает в себя не только развитие конкурентных, частных сетевых сервисов, но и разработку коммерческих продуктов, реализующих Интернет-технологию. В начале 1980-х годов десятки производителей, предвидя спрос на подобные сетевые решения, встраивали TCP/IP в свои продукты. К сожалению, они не располагали достоверной информацией о том, как Интернет-технология должна была работать, и как потенциальные покупатели предполагали использовать сети. Большинство производителей видели в TCP/IP небольшую добавку к собственным закрытым сетевым решениям. Министерство обороны США во многих контрактах требовало обязательного использования TCP/IP, но практически не помогало своим подрядчикам понять, как строить полезные TCP/IP-продукты.

Потребовалось несколько лет для проведения конференций, учебных курсов, встреч и семинаров проектировщиков, чтобы доказать преимущества новой сетевой технологии. В сентябре 1988 года состоялась первая торговая выставка Interop. В ней приняли участие 50 компаний. Выставку посетило около 5 тысяч инженеров из организаций — потенциальных клиентов. С тех пор размах торговых

выставок Interop увеличился в огромной степени. Параллельно с действиями по коммерциализации, связанными с Interop, производители начали посещать семинары, происходящие 3 или 4 раза в год, чтобы обсудить новые идеи по расширению семейства протоколов TCP/IP. Раньше на такие встречи, финансировавшиеся правительством, собиралось несколько сот человек, преимущественно из академических кругов. Теперь число участников нередко превосходит тысячу, по большей части они представляют производителей и сами оплачивают организационные расходы. Такое самоорганизующееся сообщество, объединяющее все заинтересованные стороны — исследователей, пользователей и производителей, весьма эффективно развивает семейство TCP/IP в духе сотрудничества и взаимной выгоды.

Примером сотрудничества между исследовательскими и коммерческими кругами может служить сетевое управление. На заре Интернет основной упор делался на определении и реализации протоколов, обеспечивающих совместимость. С ростом сети становилось понятно, что некоторые частные решения, использовавшиеся для управления, не всегда удается применить для всей сети. В 1987 году выявилась потребность в протоколе, обеспечивающем единообразное удаленное администрирование сетевых компонентов, таких как маршрутизаторы. Для этой цели было предложено несколько протоколов, в том числе Простой протокол управления сетью (SimpleNetworkManagementProtocol, SNMP), спроектированный из соображений простоты и ставший развитием более раннего предложения SGMP (SimpleGatewayMonitoringProtocol, простой протокол мониторинга шлюзов). Кроме SNMP были предложены протоколы HEMS (High-levelEntityManagementSystem, высокоуровневая система управления объектами) и CMIP (CommonManagementInformationProtocol – общий протокол передачи управляющей информации). В наше время практически повсеместно базой сетевого управления служит SNMP.

В последние несколько лет можно наблюдать новую фазу коммерциализации. Первоначально в коммерческой деятельности участвовали преимущественно производители базовых сетевых продуктов, а также поставщики услуг, предлагающие подключение к Интернет и базовые сервисы. В наши дни Интернет-обслуживание перешло в разряд бытового, и основное внимание теперь сосредоточено на использовании этой глобальной информационной

инфраструктуры как основы других коммерческих сервисов. Данный процесс в огромной степени ускорен широким распространением и быстрым усваиванием Web-технологии, открывающей пользователям легкий доступ к информации, расположенной по всему миру. Имеются продукты, облегчающие предоставление информации, а многие из недавних технологических разработок направлены на создание все более сложных информационных сервисов на основе базовых Интернет-коммуникаций.

3.7 Подразделения, ответственные за развитие Интернет

Существуют несколько организационных подразделений, отвечающих за развитие Интернет.

Основным из них является *InternetSociety (ISOC)* – профессиональное сообщество, которое занимается вопросами роста и эволюции Интернет, как глобальной коммуникационной инфраструктуры.

Под управлением ISOC работает *InternetArchitectureBoard (IAB)*- организация, в ведении которой находится технический контроль и координация работ для Интернет. IAB координирует направление исследований и новых разработок для протокола TCP/ IP и является конечной инстанцией при определении новых стандартов Интернет.

В IAB входятдвеосновныегруппы: *Internet Engineering Task Force (IETF) и Internet Research Task Force (IRTF)*. IETF – это инженерная группа, которая занимается решением ближайших технических проблем Интернет. В свою очередь IRTF координирует долгосрочные проекты по протоколам TCP/ IP.

Для наделения региональной сети конкретным IP – адресом существует специальное подразделение Интернет – *InternetNetworkCenter, InterNIC*.

Недавнее создание и широкое распространение Всемирной паутины привлекло в Интернет массу новых людей, никогда не причислявших себя к числу исследователей и разработчиков сетей. Была создана новая координирующая организация, *W3-консорциум (WorldWideWebConsortium, W3C)*. Новый орган принял на себя обязанности по развитию протоколов и стандартов, ассоциированных с Web.

3.8 Протокол TCP/ IP и его основные свойства

Основойсети Интернет является стек проколов *TCP/ IP (TransmissionControlProtocol/ InternetProtocol)*.

Основными преимуществами протокола TCP/ IP являются:

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

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

1 – Прикладной уровень;

2 – Основной (транспортный) уровень;

3 – Уровень межсетевого взаимодействия;

4 – Уровень сетевых интерфейсов.

Уровень межсетевого взаимодействия

Уровень межсетевого взаимодействия является стержнем всей архитектуры протокола, который реализует концепцию передачи пакетов в режиме без установления соединений, то есть дейтаграммным способом. Именно этот уровень обеспечивает возможность перемещения пакетов по сети, используя тот маршрут, который в данный момент является оптимальным. Этот уровень также называют *уровнем Интернет*, подчеркивая его основную функцию – передачу данных через составную сеть. Основным

протоколом уровня межсетевого взаимодействия является протокол IP (InternetProtocol). IP-протокол проектировался для передачи пакетов в составных сетях, состоящих из большого количества локальных сетей, поэтому он хорошо работает в сетях со сложной топологией. Так как IP-протокол является дейтаграммным протоколом, то он не гарантирует доставку пакетов до узла назначения.

Основной (транспортный) уровень

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

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

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

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

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

Прикладной уровень

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

Уровень сетевых интерфейсов

В отличие от физического и канального уровня модели OSI в архитектуре стека TCP/ IP существует несколько другая интерпретация уровня сетевых интерфейсов. Протоколы этого уровня должны обеспечить интеграцию в составную сеть локальных сетей, использующих различные технологии. Поэтому разработчики той или другой технологии должны предусмотреть возможность инкапсуляции (включения) в свои кадры IP –пакетов. Уровень сетевых интерфейсов в протоколах TCP/IP не регламентируется, но он поддерживает все популярные стандарты физического и канального уровня: Ethernet, TokenRing, FDDI, GigabitEthernet, FastEthernet и д.р. Для глобальных сетей имеется возможность работы с протоколами SLIP и PPP. Разработаны спецификации для соединения с сетями X.25, framerelay, ATM.

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

3.9 Соответствие уровней стека TCP/ IP семиуровневой модели OSI

Соответствие стека TCP/IP модели OSI показано на рис 3.1.Как видно из рисунка 3.1 протокол TCP занимает транспортный и сеансовый уровень, а на сетевом уровне используется протокол IP. Отметим, что в модели TCP/IP программные модули, соответствующие транспортному и сеансовому уровню, устанавливаются только на конечных компьютерах.

Программный модуль протокола TCP/ IP реализуется в операционной системе компьютера в виде отдельного системного модуля (драйвера). Интерфейс между прикладным уровнем и TCP представляет собой библиотеку вызовов, такую же, как, например,

библиотека системных вызовов для работы с файлами. Пользователь может самостоятельно настраивать протокол TCP/ IP для каждого конкретного случая (количество пользователей сети, пропускная способность физических линий связи и т.д.).

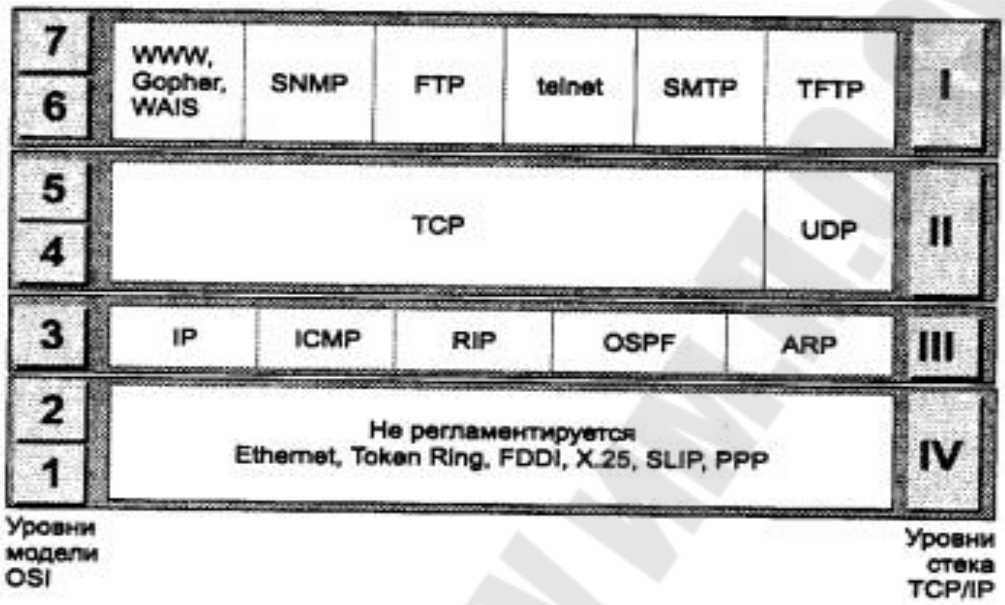


Рисунок 3.1– Соответствие стека TCP/IP модели OSI

3.10 Адресация в IP-сетях

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

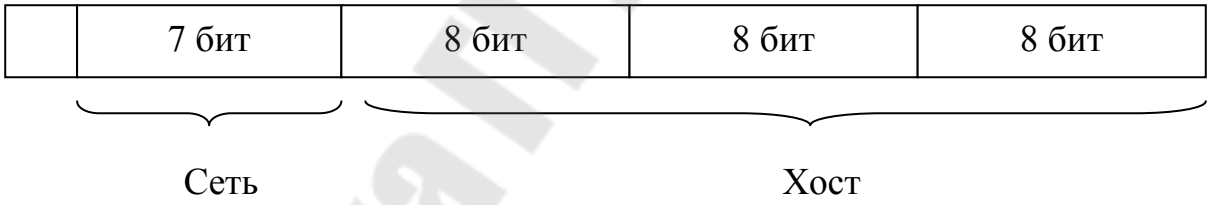
В соответствии принятым в момент разработки IP-протокола соглашением, адрес представляется четырьмя десятичными числами, разделенными точками. Каждое из этих чисел не может превышать 255 и представляет один байт четырехбайтногоIP-адреса. Выделение всего лишь четырех байт для адресации всей сети Интернет связано с тем, что в то время массового распространения локальных сетей пока не предвиделось. О персональных компьютерах и рабочих станциях вообще не было речи. В результате под IP-адрес было отведено 32

бита, из которых первые 8 бит обозначали сеть, а оставшиеся 24 бита — компьютер в сети. IP-адрес назначается администратором сети во время конфигурирования компьютеров и маршрутизаторов. Номер сети может быть выбран администратором произвольным образом, или назначен по рекомендации специального подразделения Интернет – InterNIC. Обычно поставщики услуг Интернет получают диапазоны адресов у подразделений InterNIC, а затем распределяют их среди своих абонентов. Отметим, что маршрутизатор может входить сразу в несколько сетей, поэтому каждый порт маршрутизатора имеет свой IP – адрес. Таким же образом и конечный компьютер так же может входить в несколько сетей, а значит иметь несколько IP-адресов. Таким образомIP-адрес характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение.

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

IP- адресация определяет пять классов сетей

Класс А.



Сети класса А предназначены главным образом для использования крупными организациями, их адрес начинается с 0 в двоичной записи, или с 1 в десятичной записи, они имеют номера от 1 до 126 (если все семь бит равны «1» = 1111111= 127, номер сети 0 не используется, а номер 127 используется для специальных целей). В сетях класса А предусмотрено большое количество узлов – 2^{24} = 16 777 216 узлов.

Пример.

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

00000001. 00000000. 00000000. 00000001 = 1.0.0.1

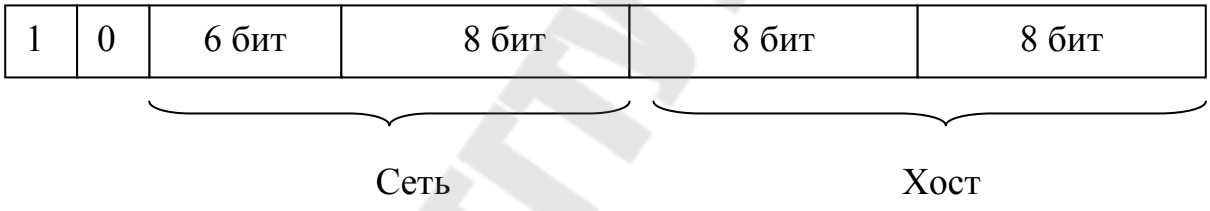
1 0 0 1

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

01111110. 11111111.11111111.11111110 = 126.255.255.254

126 255 255 254

Класс В.



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

Пример.

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

10000000.00000000.00000000. 00000001 = 128.0.0.1

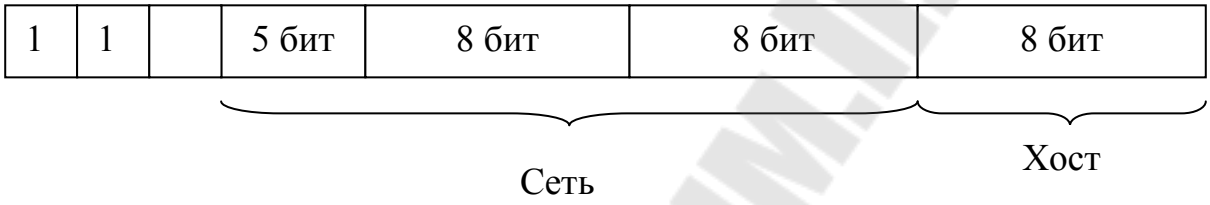
128 0 0 1

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

10111111. 11111111. 11111111. 11111110 = 191.255.255.254

191 255 255 254

Класс C.



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

Пример.

Узел имеет минимально возможный номер в сети классаCc минимально возможным номером сети

11000000. 00000000. 00000000. 00000001 = 192.0.0.1

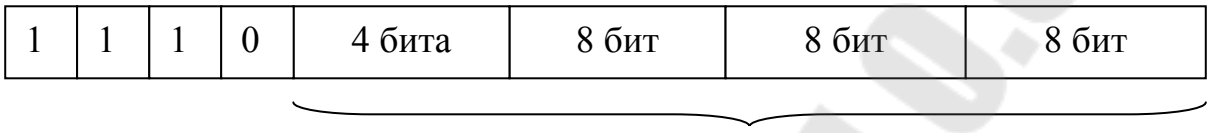
192 0 0 1

Узел имеет максимально возможный номер в сети классаCc максимально возможным номером сети

11011111. 1111111. 1111111. 11111110 = 223.255.255.254

223 255 255 254

Класс D



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

Если в пакете указан адрес сети класса D, то его получат все узлы этой сети. Поэтому сети класса D называются сетями multicast – широковещательными сетями и используются для обращения к группам узлов. Основное назначение multicast – распространение информации по схеме «один- ко- многим». Групповая адресация предназначена для экономичного распространения в Интернет или большой корпоративной сети аудио- или видеопрограмм, предназначенных сразу большой аудитории слушателей или зрителей.

Класс E

Адреса сетей класса E начинаются с 11110 в двоичной записи, или с 240 в десятичной записи, они имеют номера от 240.0.0.0 до 247.255.255.255 (11110000.00000000.00000000.00000000.=240.0.0.0,

11110111.11111111.11111111.1111111= 247.255.255.255). Сети класса Е зарезервированы для будущих использований.

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

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

3.11 Использование масок в IP-адресации

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

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

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

Маска – это число, которое используется в паре с IP- адресом; двоичная запись маски содержит единицы в тех разрядах, которые должны в IP- адресах интерпретироваться как номер сети. Единицы в маске должны представлять непрерывную последовательность.

Для стандартных классов маски имеют следующие значения:

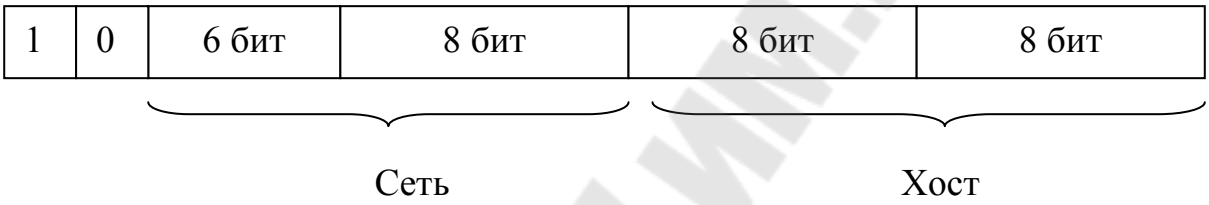
- Класс А – 11111111.00000000.00000000.00000000 (255.0.0.0)
- Класс В – 11111111.11111111.00000000.00000000 (255.255.0.0)

• Класс C – 11111111.11111111.11111111.00000000 (255.255.255.0)

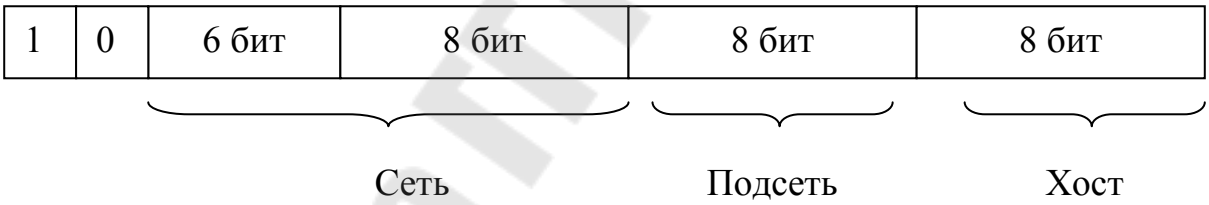
Рассмотрим, каким образом маска преобразует IP-адреса.

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

Адрес до преобразования выглядел следующим образом:



После организации подсети IP- адрес стал выглядеть:



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

Пример

IP- адрес сети класса B задан в виде:

10000010. 00100000. 10000101. 00000001 = 130.32.133.1

130 32 133 1

а) Маска не используется. В этом случае номером сети являются первые два байта и определяют сеть 130.32.0.0, а номер узла равен 0.0.133.1

б) Используется маска:
11111111.11111111.10000000.00000000 = 255.255.128.0

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

10000010. 00100000. 10000000. 00000000 = 130.32.128.0

Номер узла в этой сети становится 0.0.5.1

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

Пример

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

В двоичной форме маска имеет вид:

11111111. 11111111. 11111111. 11000000

255 255 255 192

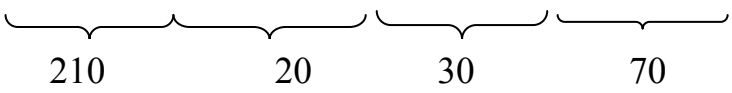
Двоичный адрес первого компьютера:

11010010. 00010100. 00011110. 11000001

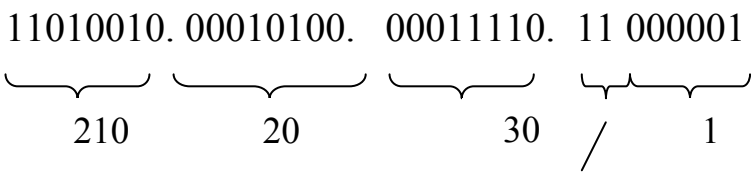
210 20 30 193

Двоичный адрес второго компьютера:

11010010. 00010100. 00011110. 01000110

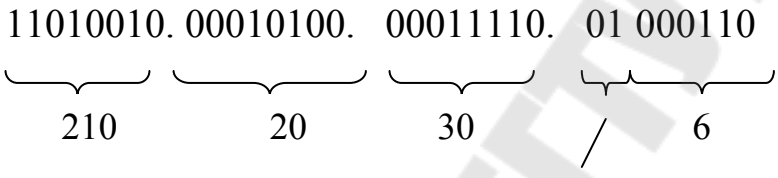


Накладывая маску на адрес первого компьютера, получим его новый адрес:



Подсеть
№ 3

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



Подсеть
№ 1

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

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

172.31.0.0, в классе С – это диапазон из 255 сетей – 192.168.0.0 – 192.168.255.0.

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

3.12 Протокол IPv6, как развитие транспортных средств IP-протокола

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

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

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

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

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

Среди других новых свойств IPv6-протокола можно отметить также более рациональную структуру формата заголовка пакета, увеличение производительности маршрутизаторов, работающих с этим протоколом, возможность маркировки потока данных, если их

необходимо обрабатывать особым образом, аутентификацию дейтаграмм и др.

3.13 Чтение таблицы маршрутизации

Хотя концепция IP маршрутизации проста, детали иногда трудно понять. Нет никакой возможности, чтобы каждый компьютер знал местоположение всех остальных компьютеров в мире. Однако Интернет работает на таких принципах, которые дают компьютеру возможность достичь любого другого компьютера, который ему нужен. Вы можете быстро подключиться к удаленному компьютеру, даже еслине знаете, где он находится. Суть решения: *любое сетевое подключение имеет очень ограниченное количество возможных действий с получаемым пакетом. Оно может игнорировать его, принять его или пропустить далее.* Это все.

Вариант «пропустить его далее» возможен, когда указан Адрес Шлюза. Когда машина решает, что она должна пропустить пакет дальше, она отправляет его по Адресу Шлюза. Очень мало сетей существуют самостоятельно, без доступа *к* или *из* внешних компьютеров. Таким образом, большинство сетей имеют компьютер с более чем одним сетевым подключением: помимо локальной сети, он подключен к другой сети. Этот компьютер определяет логическое местоположение Шлюза.В некотором смысле весь Интернет – это не что иное, как серия индивидуальных сетей, каждая с одним или более Адресом Шлюза. Когда вы подключаетесь к веб-серверу, ваши пакеты могут пройти через множество других сетей. Все эти сети должны знать, куда отправить пакеты, которые не приняты и не проигнорированы.

При настройке WinProxy в вашей локальной сети, IP адрес машины WinProxy становится Адресом Шлюза для каждой клиентской машины сети. После этого, когда приложение на машине клиента отправляет пакеты наружу, оно отправляет их, используя стек TCP/IPна этой машине. Если не уставлены другие правила, которые говорят, куда нужно отправлять пакеты, стек отправляет пакеты на машину WinProxy.

Каждая машина с TCP/IP имеет таблицу маршрутизации, то есть серию правил, которые говорят стеку TCP/IP, что делать с каждым пакетом. Эти пакеты должны прийти из сети или они должны

исходить из локального приложения для отправки через сеть. Таблица маршрутизации является удобочитаемой, поэтому вы можете взглянуть и посмотреть, какие правила использует ваша машина для управления пакетами. Чтобы посмотреть таблицу маршрутизации машины, откройте командную строку и введите командуrouteprint.

Ниже вы увидите таблицу маршрутизации машины с Windows 95. Она имеет сетевую карту с IP адресом 90.0.0.1 и маской подсети 255.255.255.0. WinProxy установлена на машине (вы не можете увидеть этого из таблицы маршрутизации) и машина на этот момент не подключена к Интернету (вы можете увидеть это из таблицы 3.1).

Каждая строка составляет правило маршрутизации. Когда стек TCP/IPрешает, куда отправить пакеты, он просматривает таблицу маршрутизации и использует следующие приоритеты:

1. Проверяет возможность TCP/IPподключения.
2. Если есть точное совпадение IP адреса, использует это правило. Если нет, то:
3. Если есть совпадение адреса сети, использует это правило. Если нет, то:
4. Если нет совпадений, использует Шлюз по умолчанию.

Таблица 3.1 – Таблица маршрутизации (без подключения к Интернету)

Сетевой адрес	**Маска сети**	**Адрес шлюза**	**Интерфейс**	**Метрика**
90.0.0.0	255.255.255.0	90.0.0.1	90.0.0.1	1
90.0.0.1	255.255.255.255	127.0.0.1	127.0.0.1	1
90.255.255.255	255.255.255.255	90.0.0.1	90.0.0.1	1
127.0.0.0	255.0.0.0	255.0.0.1	255.0.0.1	1
224.0.0.0	224.0.0.0	90.0.0.1	90.0.0.1	1
255.255.255.255	255.255.255.255	90.0.0.1	0.0.0.10	1

Несколько слов о том, что означает каждый столбец:

1. **Сетевой адрес** проверяется на совпадение с адресом получателя в IP заголовке пакета. Элементами этого столбца могут быть отдельные адреса, сетевые адреса, или шлюзы. Допустим, пакет

отправлен на адрес 90.0.0.3. Первая проверка посмотреть, есть ли 90.0.0.3 в столбце сетевого адреса таблицы. Если есть, то это точное совпадение отдельного сетевого подключения, и стек TCP/IPиспользует остальную часть линии, чтобы определить, что делать с этим пакетом. Если нет, он посмотрит, есть ли совпадение адреса сети (не смущайтесь этим неудачным двойным использованием термина «сетевой адрес»). Имеется ввиду часть 90.0.0. Например, имеется запись для адреса 90.0.0.0. Следовательно, она является правилом для пакета, который адресован сети 90.0.0.x и который не имеет точное совпадение в таблице.

2. **Маска сети** используется почти также, как и Маска Подсети, хотя это не та же самая вещь. Она говорит вам, какая часть сетевого адреса должна совпадать.

3. **Адрес шлюза** это куда пакеты будут отправлены при выборе этого правила.

4. **Интерфейс** это сетевое подключение, которое надо использовать при отправке по этому адресу.

5. **Метрика** число переходов (путешествий от одного компьютера к другому) для выполнения правила. Если случится, что два правила будут совпадать, тогда будет выбрано правило с меньшим количеством переходов. Метрика становится весьма важным параметром для больших Интернет маршрутов, и менее важным в маленьких локальных сетях.

Давайте теперь посмотрим на отдельные записи. В таблице маршрутизации перечислены три отдельных адреса (загляните в столбец Маска сети, где сетевая маска 255.255.255.255 означает, что «каждый отдельный знак сетевого адреса должен быть рассмотрен для совпадения» т.е. это индивидуальный адрес). 90.0.0.1 – это адрес сетевой карты этой машины. 90.255.255.255 это специальный адрес, используемый для широковещательных запросов к 90.x.x.x сети. 255.255.255.255 это широковещательный запрос специально-ограниченного назначения. Ни один из последних двух практически не используются.

В списке три сетевых адреса: 90.0.0.0, 127.0.0.0 и 224.0.0.0. Первый – это сеть, к которому относится 90.0.0.1; второй адрес специального назначения для локального возврата (в частности, адрес 127.0.0.1 определен как локальный внутренний адрес и означает «машина, находящаяся здесь». При использовании на любой машине, он всегда означает «эта машина здесь, на которой я сейчас работаю».

Последний, 224.0.0.0, является зарезервированным числом для мульти-кастинга.

Как машина использует их? Нам интересны только три элемента. Первый – это индивидуальный адрес, 90.0.0.1. Адрес шлюза 127.0.0.1 говорит вам, что любой пакет с таким получателем предназначен этой машине, прямо здесь, прямо сейчас. Любой пакет, прибывающий с таким адресом доступен для приложений на этой машине.

Адрес локальной сети 90.0.0. Пакет приложения локальной машины, адресованный любому адресу группы 90.0.0.x (за исключением 90.0.0.1) передается сетевой карте. Пакеты сети с одним из этих адресов (пришедшие через адрес 90.0.0.1) игнорируются.

Способ, которым записан адрес 127.0.0.0 с маской сети 255.0.0.0, подразумевает, что эта машина ответит на любой адрес в диапазоне 127.x.x.x, не только на внутренний локальный адрес 127.0.0.1. Если вы попробуете, вы убедитесь, что так и есть.

Последний кусочек информации о том, чего нет в таблице маршрутизации. Там нет адреса шлюза, который может сбивать с толку, так как есть колонка Адрес Шлюза; однако, ни одна запись в таблице не говорит компьютеру, что делать, когда другие правила потерпят неудачу. Так как эта машина имеет только один сетевой адрес, и нет других способов доступа в другую сеть, то нет нужды для правила шлюза. Следующая таблица 3.2 покажет вам, на что похоже правило шлюзапосле того как машина WinProxy подключилась к Интернету.

Таблица 3.2 – Таблица маршрутизации с подключением к Интернету

Сетевой адрес	**Маска сети**	**Адрес Шлюза**	**Интерфейс**	**Метрика**
0.0.0.0	**0.0.0.0**	**207.21.140.5**	**207.21.140.5**	**1**
90.0.0.0	*255.255.255.0*	*90.0.0.1*	*90.0.0.1*	*2*
90.0.0.1	*255.255.255.255*	*127.0.0.1*	*127.0.0.1*	*1*
90.255.255.255	*255.255.255.255*	*90.0.0.1*	*90.0.0.1*	*1*
127.0.0.0	*255.0.0.0*	*127.0.0.1*	*127.0.0.1*	*1*
207.21.140.0	**255.255.255.0**	**207.21.140.5**	**207.21.140.5**	**1**

207.21.140.5	**255.255.255.255**	**127.0.0.1**	**127.0.0.1**	**1**
207.21.140.255	**255.255.255.255**	**207.21.140.5**	**207.21.140.5**	**1**
224.0.0.0	224.0.0.0	207.21.140.5	207.21.140.5	1
224.0.0.0	*224.0.0.0*	*90.0.0.1*	*90.0.0.1*	*1*
255.255.255.255	*255.255.255.255*	*207.21.140.5*	*207.21.140.5*	*1*

Windows перезаписывает эту таблицу маршрутизации после каждого модемного подключения. Мы добавили несколько записей, чтобы помочь вам понять, что происходит. Записи, которые остались неизменными с момента, когда подключение отсутствовало, выделены курсивом. Записи с нормальным шрифтом новые специальные записи, которые мы пока проигнорируем, так как они не повлияют на нормальные операции. Новые записи, которые нас действительно интересуют, выделены жирным шрифтом.

Как вы можете видеть, эта машина теперь имеет два сетевых адреса, 90.0.0.1 и 207.21.140.5. Видите элементы 127.0.0.1 в столбце Адрес Шлюза? Этот номер, как вы знаете, означает локальный возврат, говорящий «этот компьютер, прямо здесь». Записи в столбце «адрес шлюза» это куда компьютер отправит пакет, который будет удовлетворять правилу. Пакет, адресованный 90.0.0.1, точно удовлетворяет; смотрите столбец шлюза, чтобы увидеть, что делать с этим пакетом; видите 127.0.0.1, значит это для моего компьютера.

Или в обратном порядке. Просмотрите столбец шлюза, ища знак «Вы Здесь» – адрес локального возврата. Каждый раз, когда вы видите магический адрес возврата, посмотрите в столбец сетевого адреса, и вы увидите 90.0.0.1 и 207.21.140.5.

Так как этого второго адреса не было здесь раньше, вы знаете, что этот динамический адрес был назначен вам, когда вы подключились к провайдеру. Есть несколько новых простых правил, один важнее другого. Простые правила: все, что адресовано сети 90.0.0.0 направляется сетевому подключению 90.0.0.1, и все адресованное сети 207.21.140.0 направляется сетевому подключению 207.21.140.5. Это достаточно просто. Тем не менее, самая первая строка немного меняет поведение. Сетевой адрес 0.0.0.0 переводится примерно как «любой адрес». Это правило шлюза. Если адрес назначения пакета не удовлетворяет индивидуальному адресу в таблице, и не удовлетворяет сетевому адресу, он все еще должен

удовлетворять этому правилу. Любой адрес, не определенный иначе, будет отправлен сетевому подключению 207.21.140.5.

Так, когда WinProxy отправляет пакет машине, скажем, 188.3.2.1, стек посылает пакет сетевому подключению 207.21.140.5. Этот адрес часть сети вашего провайдера, и где-то в этой сети есть другая машина с адресом шлюза, так что пакет просто продолжит свой путь от шлюза к шлюзу, пока не достигнет указанного адресата.

Эта информация полезна потому, что, по крайней мере, она иллюстрирует важность DNS. Как вы можете видеть, в таблице маршрутизации нет имен; там нет даже места для имен. Когда одно из ваших приложений использует имя (например, http://www.excite.com), оно сперва должно быть сконвертировано в IP адрес, перед тем как пакеты можно будет отправлять.

Другая вещь, которую вы заметите, когда компьютер является частью двух различных сетей (как ваш компьютер WinProxy), должно быть ясное различие между двумя сетями. Если различия не будет, стек TCP/IPбудет отправлять пакеты в произвольном направлении. Вы не можете иметь два адреса на машине WinProxy, являющиеся частью одной сети.

Что еще вы ищете? Возможно, ваша локальная сеть является подмножеством сети провайдера, отличающейся только другой маской подсети. Если так и есть, вам нужно изменить одно или другое.

Большую часть времени, люди с простой, единственной локальной сетью не нуждаются в том, чтобы смотреть в таблицу маршрутизации, чтобы помочь устранить проблемы подключения. Она чаще используется теми, кто наблюдает за множеством сетей. Если это ваша ситуация, изучите таблицу маршрутизации, чтобы убедиться, что пакеты имеют маршрут в Интернет и что возвращающиеся пакеты смогут однозначно вернуться к исходному компьютеру.

3.14 Система доменов DNS

Выше было установлено, что для обращения к хостам используются 32-разрядные IP-адреса. Поскольку при работе в сети Интернет использовать цифровую адресацию сетей крайне неудобно, то вместо цифр используются символьные имена, называемыми *доменными именами.* Доменом называется группа компьютеров,

объединенных одним именем. Символьные имена дают пользователю возможность лучше ориентироваться в Интернет, поскольку запомнить имя всегда проще, чем цифровой адрес.

На заре создания Интернет соответствия между именами хостов и их IP-адресами были размещены в единственном файле, который назывался Hosts.txt, который размещался на компьютере в центре InterNIC. Этот файл передавался по всем хостам еще совсем тогда крохотной сети. Стремительный рост Интернет заставил выработать новую концепцию механизма разрешения имен. С этой целью была разработана специальная система DNS (DomainNameSystem), для реализации которой был создан специальный сетевой протокол DNS. Начальные попытки создать единую копию целой базы данных имен и адресов оказались тщетными из-за громадного объема информации. Было принято решение строить распределенную базу данных, а для увеличения производительности использовать механизм локального кэширования (сохранения в локальной базе данных). Доступ к распределенной базе данных не зависит ни от аппаратной платформы хоста, ни от коммутационной системы. Доступ к базе данных должны иметь все пользователи Интернет. Администрирование базы данных DNS возлагается на каждую организацию, которая подключается к Интернет. Организация должна инсталлировать свой собственный компьютер – сервер разрешения имен и ту часть распределенной базы данных, содержащей информацию о домене хостов данной организации. Сервер должен обслуживать хосты внутри организации и предоставлять доступ к базе данных этой организации извне.

Структура баз данных в системе DNS имеет иерархический вид, аналогичный иерархии файлов, принятой во многих файловых системах. Дерево имен начинается с корня, затем следует старшая символьная часть имени, вторая часть имени и т.д. Младшая часть имени соответствует конечному узлу сети. Все имена разделяются точками, причем иерархия задается справа налево, например, www.bseu.minsk.by

По имени можно получить информацию о профиле организации или ее местоположении. Шесть доменов высшего уровня определены следующим образом:

- gov – правительственные организации;
- mil – военные организации;
- edu – образовательные организации;
- com – коммерческие организации;

- org- общественные организации;
- net – организации, предоставляющие сетевые услуги, как правило, региональные сетевые организации.

Кроме того, все страны мира имеют свое собственное символьное имя, обозначающий домен верхнего уровня этой страны. Например, de – Германия, us – США, ru- Россия, by – Беларусь и т.д. Таким образом, адрес www.cdo.bseu.minsk.by означает, что компьютер дистанционного образования cdo находится в группе компьютеров (в домене) Белорусского государственного экономического университета bseu, в домене minsk в Республике Беларусь. Графически DNS можно представить в виде дерева, как на рисунке 3.2.

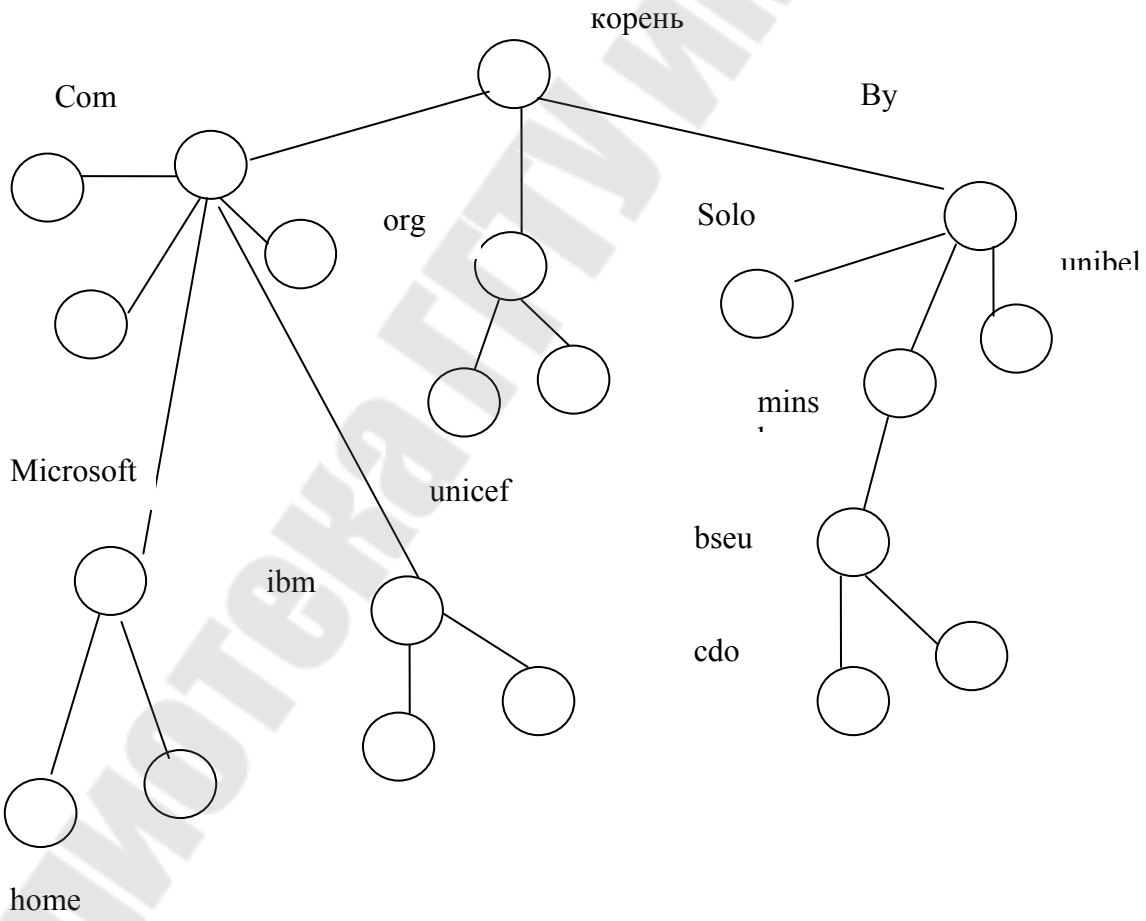


Рисунок 3.2– Дерево системы DNS

DNS имеет три основные компоненты:

• Пространство имен домена (domainnamespace) и записи базы данных DNS (resourcerecords).Они определяют структуру имен «дерева» и данных, связанных с этими именами. Запрос по данному имени возвратит IP- адрес хоста.

• Сервера имен (nameservers). Сервера имен – это специальные компьютеры со специальными серверными программами, обрабатывающие информацию имен и данных имен. Сервер управляет всей информацией подчиненной ему области имен и данных домена. При обращении за информацией, который данный сервер не обслуживает, он должен или переправить запрос серверу, обслуживающему эту информацию, или стоящему на следующей ступени иерархии. Сервер, в распоряжении которого находится определенная часть информации об именах, является владельцем (authority) имен домена, а граница владения называется зоной (zone). Зоны строятся не на основе принадлежности какой-либо части данных к определенной организации, а распределяются автоматически серверами имен и должны обеспечить полную адресацию хостов.

• Программы разрешения имен (resolves). Эти программы возвращают информацию, хранящуюся в базе данных имен домена по запросу пользователя. Пользователь взаимодействует с пространством имен через указанные программы. Как правило эти программы реализуются в виде системного модуля, напрямую связанного с пользовательской программой, поэтому не требуется ни какого дополнительного протокола обмена.

Основным предназначением системы имен доменов является обеспечение механизма именования ресурсов. Этот механизм должен эффективно работать с различными хостами, сетями, семействами протоколов и типами организаций. Описанная выше структура DNS позволяет решать проблему адресации отдельных модулей изолировано, и, тем самым, создает универсальную модульную архитектуру.

Пользователь взаимодействует с пространством имен через программы разрешения. Для работы программ разрешения необходимо обращаться к серверам имен на других хостах, что может давать задержки от миллисекунд до нескольких секунд. Поэтому одной из важнейших свойств программ разрешения имен является возможность устранения сетевых задержек ответов. При этом используется механизм *кэширования результатов запросов имен.*

Этот механизм ускоряет процесс определения имен, так в КЭШ-памяти накапливается информация о всех предыдущих именах , к которым обращалась программа.

Наиболее упрощенный и распространенный принцип работы такой программы с серверами имен показан на рисунке 3.3.

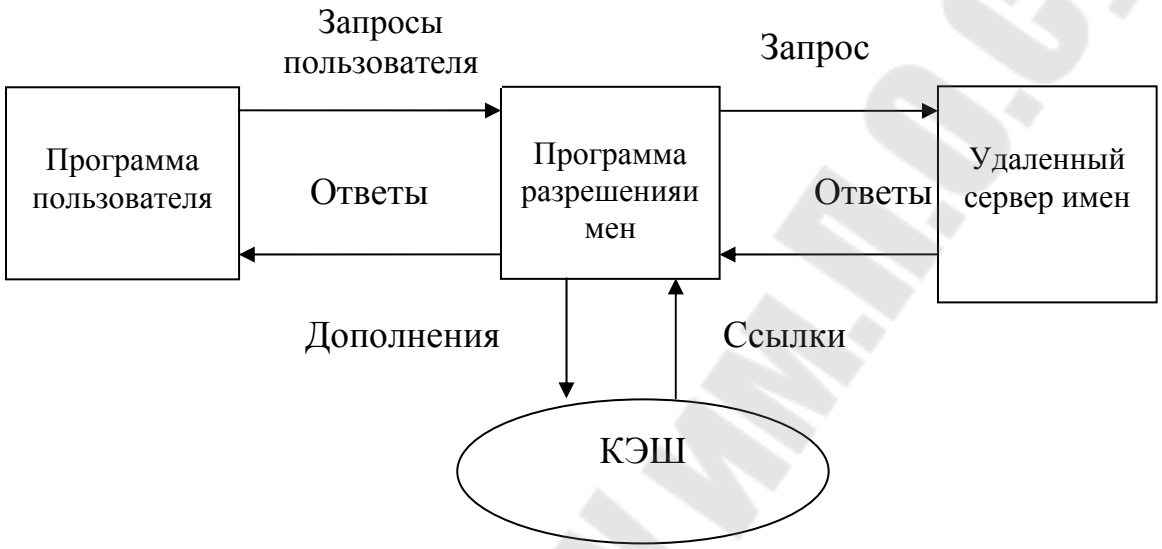


Рисунок 3.3 – Принцип работы с серверами имен

Программа пользователя запрашивает имя хоста и передает этот запрос программе разрешения имен. В первую очередь программа разрешения имен обращается за необходимым IP- адресом в собственную КЭШ- память. Если требуемого имени а КЭШ – памяти не находится, программа разрешения имен обращается к удаленному серверу имен, В случае нахождения необходимого имени, программа возвращает пользователю требуемый IP- адрес, одновременно записывая его в КЭШ- память.

Система DNS требует, чтобы доступ к информации определенной зоны мог быть осуществлен с нескольких серверов доменов. Существует механизм предоставления пользователям различных доменов совместного использования информации путем установления *доверительных отношений* между доменами. При этом доверительные отношения могут быть как *двухсторонними,* так и *односторонними*.

При двухсторонних доверительных отношениях пользователь любого из двух доменов имеет доступ к информации, находящихся на соседнем домене.

При односторонних доверительных отношениях пользователь, находящийся в доверяемом домене, имеет доступ к серверам домена-доверителя, но не наоборот.

3.15 Протоколы прикладного уровня

3.15.1 Протоколы электронной почты

Электронная почта (E-mail) – позволяет обмениваться сообщениями с пользователями на других компьютерах. Для обеспечения этого сервиса созданы специальные почтовые серверы, которые хранят сообщения для своих пользователей. Чтобы прочитать полученные сообщения, пользователю необходимо присоединится к такому серверу и работать со своими сообщениями как с отдельными файлами. Скорость, эффективность и простота использования сделали этот сервис наиболее распространенным в Интернет. Работу с электронной почтой осуществляют разные протоколы, наиболее популярными из которых являются SMTP, POP3 и IMAP4.

Протокол SMTP

Основным протоколом работы с электронной почты является SMTP (SimpleMailTransferProtocol – простой протокол передачи почты). Протокол SMTP поддерживает передачу сообщений электронной почты между произвольными узлами Интернет. Он служит для достоверной и надежной передачи сообщений между хостами. Существует большое множество почтовых программ, использующих этот протокол: OutlookExpress, MicrosoftMail, Lotus и т.д.

Протокол SMTP представляет собой независимый от транспортной подсистемы протокол для работы которого необходим только транспортный канал передачи потока данных. SMTP может работать по любому транспортному каналу, удовлетворяющему требованиям передачи данных через сети или группы сетей, например, TCP, X.25 ид.р.

Протокол SMTP обеспечивает как передачу сообщений в адрес одного получателя, так и тиражирование нескольких копий

сообщений для передачи в разные адреса. Протокол SMTP может передавать не только текстовые сообщения, но и рисунки, исполняемые файлы и т.д. Основными составляющими заголовка протокола являются From, To, Date, Subject, Message – ID. При передаче сообщения через промежуточные почтовые сервера к заголовку прибавляются записи Received, содержащие поля с адресами и временем обработки сообщений промежуточными серверами. Для более удобной работы с составными и нестандартными сообщениями (графика, видео) был разработан новый формат упаковки почтовых сообщений – MIME (MultipurposeInternetMailExtension, многоцелевое расширение электронной почты), в котором в заголовок протокола добавляются дополнительные поля.

Схема работы SMTP выглядит следующим образом:

1. Отправитель посылает команду (MAIL), идентифицирующую атрибуты отправителя почты, например, его адрес. Если получатель может принять почтовое сообщение, он отправляет в ответ команду ОК.
2. После этого отправитель отправляет команду (RCTP), идентифицирующую атрибуты получателя почты, например, адрес почтового ящика. Если получатель готов принять почту в данный ящик, он отвечает командой ОК, если нет, он отвечает отказом принять почту в указанный почтовый ящик.
3. Отправитель отправляет данные получателю. Если получатель успешно принял данные, отправляет команду ОК.

Протокол SMTP поддерживает несколько механизмов передачи почты, основными из которых являются: напрямую от хоста отправителя к хосту пользователя, когда два хоста соединены между собой напрямую; через серверы SMTP – хосты посредники, когда отправитель и получатель не могут соединиться напрямую.

Отметим, что очень редко удается отправить почтовое сообщение адресату напрямую. Как правило, используются SMTP-серверы, которые исполняют роль промежуточных пунктов пересылки сообщений. SMTP- серверы принимают всю поступившую почту и затем, самостоятельно, переправляют ее адресату. Этот процесс называется *ретрансляцией сообщений*. SMTP-серверы выбирают путь сообщения по своему усмотрению в зависимости от параметров настройки, скорости доступа и т.д. Протокол SMTP также позволяет отправителю самостоятельно указывать путь

передачи сообщения, устанавливая в качестве параметров команды отправки промежуточные SMTP-серверы.

SMTP-протокол использует TCP как транспортный протокол, который обеспечивает достоверность и надежность доставки сообщения. По умолчанию TCP-протокол подключен к протоколу SMTP через порт 25.SMTP-команды инкапсулируются в поле данных TCP в соответствии с обычными механизмами инкапсуляции стека протоколов TCP/ IP.

Протокол POP 3

Для небольших организаций невыгодно держать у себя систему для передачи сообщений. Это связано с тем, что в небольших организациях рабочие станции клиентов не имеют достаточных вычислительных ресурсов для обеспечения работы полного SMTP-протокола. Кроме того, таким пользователям электронной почты невыгодно держать персональный компьютер постоянно подключенным к Интернет.

Для решения этой проблемы был разработан почтовый протокол для работы в офисе – POP (PostOfficeProtocol). Его наиболее распространенный вариант- POP3.

POP3 – это простейший протокол для работы пользователя для работы со своим почтовым ящиком. Он позволяет только забрать почту из почтового ящика сервера на компьютер клиента и удалить ее из почтового ящика на сервере.

POP3-сервер не отвечает за отправку почты, он работает только как универсальный почтовый ящик для группы пользователей.Когда необходимо отправить сообщение, он должен установить соединение с каким-либо SMTP-сервером и отправить туда свое сообщение. Этот SMTP-сервер может располагаться на том же хосте, где работает POP3- сервер, а может располагаться в другом месте Интернет. Как правило, при работе с электронной почтой для получения корреспонденции используют POP3- сервер, а отправляют почту по SMTP-протоколу на один из хорошо доступных SMTP- серверов.

POP3- протокол подключается к транспортному уровню TCP через 110-й порт, который будет находится в режиме ожидания входящего соединения. После установления соединения клиент и сервер начинают обмениваться командами и данными. После окончания обмена POP3- канал закрывается.

Простота протокола, которая послужила росту его популярности вначале, обернулась затем отсутствием гибкости и невозможности

выполнять другие операции управления почтовыми ящиками. На смену протоколу POP3 пришло новое поколение протоколов работы с электронной почтой – протоколы IMAP.

Протокол IMAP4

Протокол IMAP4 (InternetMessageAccessProtocol, Version4, протокол доступа к электронной почте Интернет, версия 4) позволяет клиентам получать доступ и манипулировать сообщениями электронной почты на сервере. Существенным отличием протокола IMAP4 от протокола POP3 является то, что IMAP4 поддерживает работу с системой каталогов (или папок) сообщений. IMAP4 позволяет управлять каталогами (папками) удаленных сообщений так же, как если бы они располагались на локальном компьютере.IMAP4 позволяет клиенту создавать, удалять и переименовывать почтовые ящики, проверять наличие новых сообщений и удалять старые. Благодаря тому, что IMAP4 поддерживает механизм уникальной идентификации каждого сообщения в почтовой папке клиента, он позволяет читать из почтового ящика только сообщения, удовлетворяющие определенным условиям или их части, менять атрибуты сообщений и перемещать отдельные сообщения.

При работе с протоколом TCP, IMAP4 использует 143-й порт. Принцип работы протокола IMAP4 такой же, как и у других подобных протоколов. Сначала клиент и сервер обмениваются приветствиями. Затем клиент отправляет на сервер команды и данные. Сервер, соответственно, передает клиенту ответы на обработку команд и данные.После завершения обмена канал закрывается.

3.15.2 Протокол работы с WWW – HTTP

HTTP (HypertextTransferProtocol, протокол передачи гипертекста) обеспечивает высокопроизводительный механизм передачи мультимедийной информации независимо от типа представленных данных. Протокол построен по объектно-ориентированной технологии и может использоваться для решения различных задач, например, для управления распределенными информационными системами.

WWW (WorldWideWeb всемирная паутина) состоит из компьютеров, которые предоставляют графический доступ к хранящейся на них информации. Способность хранить мультимедийную информацию, такую как видео, аудио, картинки и звуки, делает WWW уникальным средством распространения

информации. WWW-сервер представляет собой компьютер, на котором работает соответствующее программное обеспечение, позволяющее пользователям Интернет подсоединяться и пользоваться WWW-ресурсами этого компьютера для поиска и выбора информации. С 1999 года протокол HTTP используется системой WWW в качестве основного протокола работы.

Сервисы WWW

Протокол HTTP позволяет получать доступ к ресурсам и сервисам WWW-серверов. Для унификации доступа WWW-серверы поддерживают комплекс интерфейсов, позволяющих структурировать уровни и методы работы с различными ресурсами сети. Для работы с WWW-серверами используются следующие основные сервисы:

- URL (UniformResourceLocator, местонахождение ресурса) – предназначен для идентификации типов, методов и компьютеров, на которых находятся определенные ресурсы, доступные через Интернет. ЭтотсервисможетиметьименаURI (UniformResourceIdentifier), URN (UniformResourceName).
- HyperTextMarkupLanguage (HTML) – это язык описания содержащейся на WWW – сервере информации. HTML-файл представляет собой обычный двоичный текст (ASCII-текст), содержащий специальные коды, которые обозначают присоединенную к файлу графику, видео, аудио информацию или исполняемые коды среды для просмотра информации – коды Webbrowser, JavaScript. Когда Webbrowser получает доступ к этому файлу, он предоставляет для пользователя всю информацию в графическом или текстовом виде Web-страницы. Основная концепция размещения информации на Web-странице – это использование гиперссылки (HyperText). Гиперссылки имеют связи внутри документа и позволяют быстро переходить от одной части документа к другой или к другому документу. Гиперссылки позволяют перемещаться также на другие WWW-серверы. Это открывает возможности навигации по сети Интернет. Совокупность взаимосвязанных друг с другом гипертекстовыми ссылками и объединенных единой темой страниц называется *Web-сайтом.*
- IDC (InternetDatabaseConnector) и ASP (ActiveServerPage) – сервисы, используемые для выборки информации из баз данных и размещения их на Web – страницах.

Принципы работы HTTP – протокола

Протокол HTTP построен по модели «запрос-ответ». В запросе клиентом указываются тип запроса, URL и содержание запроса, например, параметры клиента. Сервер HTTP отвечает строкой статуса обработки запроса, которая содержит: версию поддерживаемого протокола, код обработки запроса или код ошибки и возвращаемую по запросу информацию. В простейшем случае, соединение представляет собой дейтаграммный поток данных между клиентом и сервером. В более сложной ситуации, в процессе передачи данных принимают участие несколько промежуточных объектов: *(промежуточный агент), gateway (шлюз), tunnel (туннель).*

• *Proxy* представляет собой промежуточный агент, который принимает запрос клиента и передает запрос далее по цепочке другим серверам. В момент принятия запроса proxy может работать как сервер, а при передаче запроса – как клиент. На proxy могут создаваться копии наиболее часто запрашиваемыхWeb-страниц. В этом случае клиент получает информацию с proxy, что ускоряет работу Интернет. Как правило, proxy представляет «главные ворота» выхода пользователей из внутренней сети в Интернет. В зависимости от настроек proxy может изменять часть или все сообщение запроса.

• *Gateway*представляет собой промежуточный сервер. В отличие от proxy шлюз принимает запросы клиента и без изменения передает их далее, т.е. работа шлюза прозрачна для клиента. В обратном направлении, от сервера к клиенту, шлюз наоборот, в зависимости от настроек, может пропускать или не пропускать определенную информацию. Шлюз является «главными воротами» для входа пользователей внешней сети во внутреннюю сеть.

• *Tunnel* представляет собой программу-посредник между клиентом и сервером. Туннели используются в тех случаях, когда необходимо организовать поток данных через какой-нибудь промежуточный объект (например*proxy*), который не может интерпретировать структуру потока данных.

Отметим, что ответы серверов могут храниться в КЭШе – локальной базе данных, которая возвращает их клиенту, не передавая запрос следующему серверу.

При работе по протоколу TCP сервер HTTP, как правило, использует порт 80, хотя возможно использование и других портов.

Тенденции развития протокола HTTP:

1. Увеличение производительности за счет более эффективной работы с КЭШем, промежуточными агентами.

2. Происходит расширение возможностей передачи распределенных ресурсов

3. Развиваются дополнительные механизмы защиты передаваемых данных.

3.15.3 Протокол передачи файлов FTP

FTP (FileTransferProtocol, протокол передачи файлов) – один из первых протоколов Интернет.

FTP предназначен для разделенного доступа к файлам на удаленных хостах, прямого или косвенного использования ресурсов удаленных компьютеров, обеспечения независимости клиента от файловых систем удаленных хостов эффективной и надежной передачи данных, находящихся в файлах.

Протокол FTP поддерживает сразу два канала соединения – канал передачи команд и канал передачи данных.

Для хранения файлов используются специальные FTP–серверы.

При работе по протоколу TCP сервер HTTP, как правило, использует порт 21.

3.15.4 Протокол передачи новостей NNTP

NNTP (NetworkNewsTransportProtocol, протокол передачи новостей) предназначен для тиражирования статей в распределенной системе ведения дискуссий UseNet.UseNet состоит из нескольких групп, называемыми группами новостей. Группы новостей организованы в определенном порядке, основанном на распределении дискуссий по темам, например, отдых, спорт, новости, информация, религия и др. Внутри каждой из этих групп может быть от нескольких до тысяч групп, которые обладают своей структурой.

Группы новостей позволяют пользователям с общими интересами обмениваться интересными сообщениями, отправлять свои статьи и отвечать на заметки других пользователей. После того, как статья отправлена в группу новостей UseNet, она рассылается через сервис на другие компьютеры Интернет, где установлен сервис UseNet. Этот сервис позволяет вести дискуссии по выбранной теме, осуществлять фильтрацию статей по ключевым словам и т.д.

3.15.5 Протокол удаленного терминала TELNET

Протокол удаленного терминала TELNET предоставляет возможность работать на удаленном компьютере сети, поддерживающим сервис TELNET. Принцип работы этого сервиса заключается в том, что пользователь работает с удаленным компьютером, не замечая свой собственный. Хотя физически все

данные вводятся пользователем в свой компьютер, но попадают они в удаленный компьютер, т.е. собственный компьютер пользователя является только средством, обеспечивающим сеанс связи. Этот сервис составлял в прошлом основу работы Интернет. В настоящее время TELNET используется, в основном, для удаленного администрирования сети.

Кроме указанных выше наиболее популярных протоколов в Интернет используются и другие протоколы, такие как *сетевая файловая система (NSF), мониторинг и управление сетью (SNMP), удаленное выполнение процедур (RPC), сетевая печать и др.*

3.16 Структурные компоненты сети Интернет

Кроме рассмотренных выше важнейших структурных компонент глобальной сети Интернет, таких как маршрутизаторы, DNS – серверы, а также серверы соответствующих протоколов прикладного уровня в Интернет широко используются понятия *файрволл (firewall)*, *брандмауэр и провайедер (provider)*

Файрволл

Файрволлом называется программмно-аппаратный комплекс защищающий локальную сеть от несанкционированного доступа, например, от атак хакеров или проникновения вирусов. У пожарных так именуется стена из огнеупорного материала, предотвращающая распространение огня. В сети файерволл обеспечивает фильтрацию прохождения информации в обе стороны и блокирует несанкционированный доступ к компьютеру или локальной сети извне. Как уже указывалось выше, любое соединение в Интернете инициируется каким-либо протоколом прикладного уровня, использующим для работы свой порт, идентифицируемый номером. Файрволл позволяет контролировать использование портов и протоколов, "прятать" неиспользуемые порты для исключения атаки через них, а также запрещать/разрешать доступ конкретных приложений к конкретным IP- адресам. Другими словами, контролировать все, что может стать орудием хакера и недобросовестных фирм.

Файрволл должен сам быть неприступным для внешних атак.

В основном файрволлы работают на сетевом уровне и осуществляют фильтрацию пакетов, хотя можно организовать защиту и на прикладном или канальном уровне. Технология фильтрации

пакетов является самым дешевым способом реализации файрволла, т.к. в этом случае можно проверять пакеты различных протоколов с большой скоростью. Фильтр анализирует пакеты на сетевом уровне и не зависит от используемого приложения.

Брандмауэр

Брэндмауэр — это своего рода программныйфайрволл. Но если быть более точным, то файрволл — это непосредственно компьютер, стоящий между локальной и внешней сетью, брэндмауэр – это программное средство контроля за входящей и исходящей информацией. Программы-брэндмауэры встраиваются в стандартные операционные системы, например, в WindowsXP или могут устанавливаться на proxi сервере.

Провайдер

Провайдер –это поставщик доступа к Интернет. Другими словами – это любая организация, предоставляющая частным лицам или организациям выход в Интернет. Провайдеры вообще разделяются на два класса:

- поставщикидоступаИнтернет (Internet access providers – ISP);
- поставщики интерактивных услуг (onlineserviceproviders – OSP).

ISP может быть предприятием, которое оплачивает быстродействующее соединение с одной из компаний, являющейся частью Интернет (такие, как AT$T, Sprint или MCI в США). Это могут быть также национальные или международные компании, которые имеют их собственные сети (типа WorldNet,Белпак, ЮНИБЕЛ и д.р.)

OSP, иногда называемые просто "интерактивные услуги", также имеют собственные сети, но обеспечивают дополнительные информационные службы, не доступные для клиентов, которые не подписались на данные услуги. Например, OSPMicrosoft предлагают пользователям доступ к Интернета-сервису фирмы Microsoft, также как к AmericaOnline, IBM и нескольким другим.

ISP-провайдеры являются наиболее распространенными.

Обычно крупный провайдер имеет собственную "точку присутствия" POP (point-of-presence) в городах, где происходит подключение локальных пользователей.

Различные провайдеры для взаимодействия друг с другом договариваются о подключения к так называемым точкам доступа NAP (NetworkAccessPoints), посредством которых происходит

объединение информационных потоков сетей, принадлежащих отдельному провайдеру.

В Интернете действуют сотни крупных провайдеров, их магистральные сети связаны через NAP, что обеспечивает единое информационное пространство глобальной компьютерной сети Интернет.

В общем виде схема Интернет представлена на рисунке 3.4.

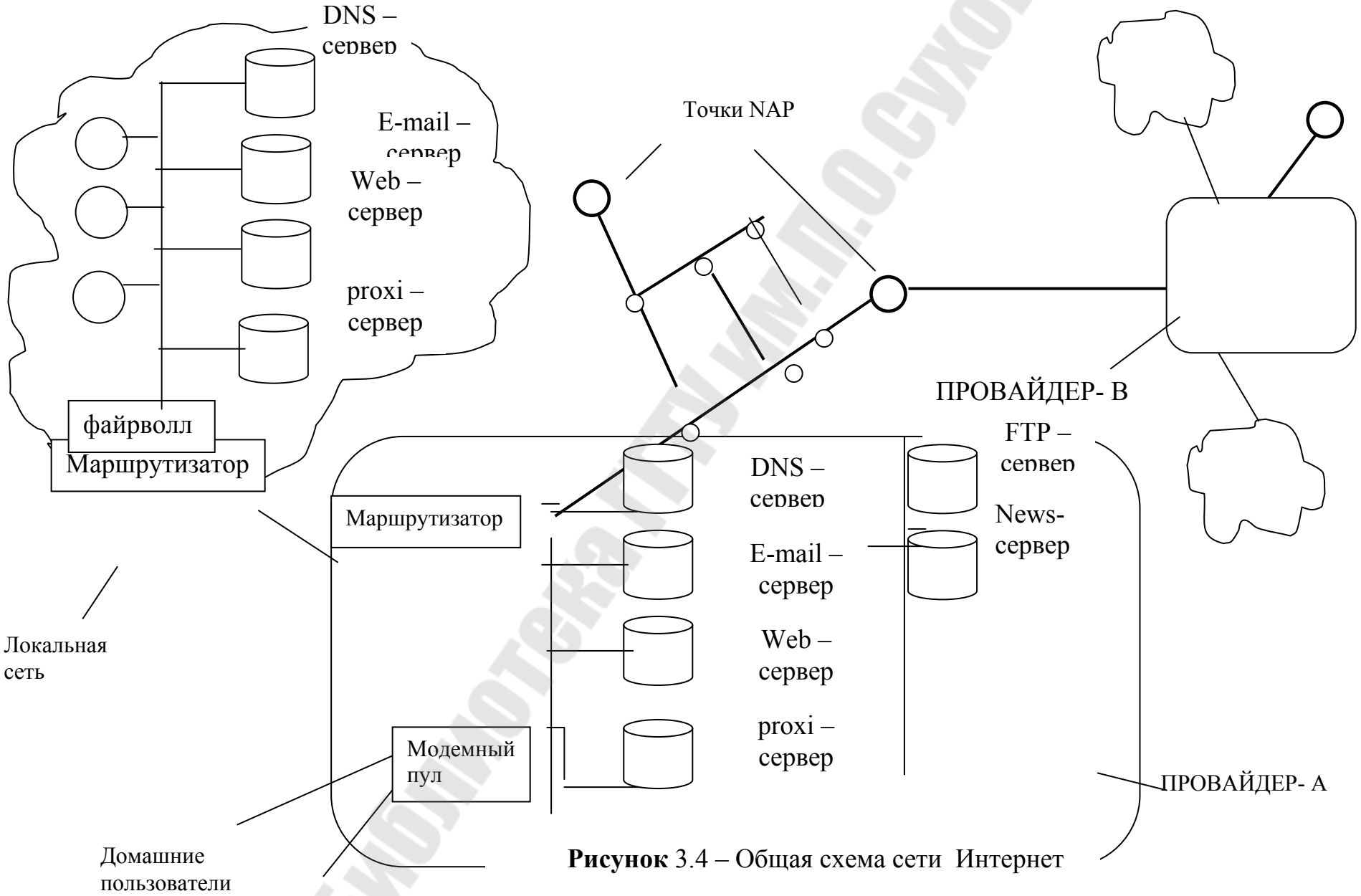


Рисунок 3.4 – Общая схема сети Интернет

3.17 Безопасность компьютерных сетей

3.17.1 Угрозы компьютерным сетям

Основным из принципов построения и функционирования глобальной сети Интернет является принцип ее доступности и открытости. Это обуславливает, с одной стороны, невероятные темпы развития сети, а с другой стороны, существенно обостряет процессы обеспечения безопасности. По самым скромным оценкам потери фирм и банком от несанкционированного доступа ежегодно составляют миллиарды долларов. Поэтому вопросы обеспечения безопасности компьютерных сетей является крайне актуальной задачей.

Отметим некоторые основные приемы нарушения безопасности компьютерной сети и противодействия этому.

Физический несанкционированный доступ

Подключение дополнительного компьютерного терминала к каналам связи путем использования шнура в момент кратковременного выхода из помещения пользователя.

Противодействие

Покидая рабочее место не оставлять персональный компьютер в активном режиме или надежно закрывать помещение.

Компьютерный абордаж (взлом системы)

Подбор пароля к системе вручную или с использование специальной программы.

Противодействие

Ограничение количества попыток неправильного ввода пароля с последующей блокировкой компьютера.

Маскарад (Мистификация)

Проникновение в сеть, выдавая себя за законного пользователя, с применением его паролей и других идентифицирующих шифров. Создание условий, когда законный пользователь осуществляет связь с нелегальным пользователем, будучи абсолютно уверенным, что он работает с необходимым ему абонентом.

Противодействие

Необходимо использовать надежные средства идентификации и аутентификации, блокирование попыток взлома системы, контроль входа в нее. Необходимо фиксировать все события в системном журнале для последующего анализа.

Сборка мусора

После окончания работы обрабатываемая информация не всегда полностью удаляется. Часть данных, оставшаяся на дисках и оперативной памяти, собирается и обрабатывается.

Противодействие

Заполнение памяти нулями или единицами, перезапись информации в другое место.

Люки

Недокументированная производителем программного обеспечения точка входа в программный модуль используется для активного воздействия на эту программу.

Противодействие

При приемке программного обеспечения необходимо производить анализ программ с целью обнаружения люка.

Троянский конь

Программа, выполняющая невидимые для пользователя действия в дополнение к выполняемым программам. В основном «ворует» и запоминает коды всех нажатых клавиш клавиатуры пользователя.

Противодействие

Создание закрытой среды использования программного обеспечения.

Вирус

Программа, которая заражает другие программы путем включения в них своих кодов, при этом зараженная программа имеет способность дальнейшего размножения. Вирусы в основном разрушают программное обеспечение.

Противодействие

Использование антивирусного программного обеспечения и специальных программ.

Червь

Распространяющаяся через сеть программа, не оставляющая своей копии на винчестере пользователя.

Противодействие

Использование антивирусного программного обеспечения и специальных программ.

«Жадные» программы

Программы, монопольно захватывающие ресурсы системы, не давая другим программам его использования.

Противодействие

Ограничение времени исполнения программ.

Кроме указанных вышеуказанных приемов нарушения безопасности используется ряд других, таких как бухинг (электронное блокирование), наблюдение, скрытые каналы и др.

Отметим, что приемы нарушения безопасности компьютерной сети делятся на так называемые конструктивные и деструктивные. При конструктивном воздействии основной целью несанкционированного доступа является получение копии конфиденциальной информации, т.е. можно говорить о разведывательном характере воздействия. При деструктивном воздействии конечной целью является разрушение информационного ресурса.

3.17.2Где может нарушаться безопасность сети

Использование протокола TCP/ IP

В протоколе TCP/ IP используется принцип ожидания получения квитанции о правильности получения сегмента информации. В случае неполучения подтверждения сервер удаленного доступа фиксирует незавершенное соединение. Программное обеспечение современных серверов может обрабатывать ограниченное количество незавершенных соединений, что приводит к фактическому блокированию доступа к серверным ресурсам при большом количестве незавершенных соединений.

Кроме этого, большинство реализаций протокола TCP/ IP имеет ограничение на количество соединений, которые могут быть установлены в единицу времени. Все соединения, превышающие это количество, не будут временно обрабатываться, и ждать своей очереди. Если посылать, например, сорок запросов на соединение в одну минуту, то сервер будет блокировать любой доступ приблизительно на десять минут. Периодическое повторение этой операции может привести к полному отключению сервера от сети.

Подмена URL – адресов для перенаправления запросов

Такая подмена становится возможной за счет перехвата и анализа «на лету» пакетов. В общем случае меняется заголовок IP-пакета и пакет перенаправляется на специальный сервер. Вся информация, проходящая через маршрутизатор, перенаправляется на этот сервер.

Электронная почта в основном взламывается путем внесения искажений в конфигурационные файлы SMTP- и POP-серверы или банальным взломом паролей доступа.

Ресурсы рассылки новостей USENET можно изменять путем несанкционированного получения привилегированного доступа к потоку новостей. В случае взлома сервера один из пользователей получает возможность отправки запросов класса slave (ведомый сервер) и тем самым заблокировать доступ пользователей системы к каналу распространения новостей, а также создать для себя лично эффективный канал снятия новостей.

Методы несанкционированного доступа к ресурсам WWW ориентированы в основном на нештатное использование HTML-страниц программ-браузеров, а также на применение программ, расширяющих функциональные возможности браузеров для решения несвойственных им задач.

Одним из наиболее эффективных путей снижения эффективности работы сети состоит в увеличении бесполезного или даже вредного, с точки зрения решаемых задач, сетевого трафика. Этот метод основан на особенности психологии людей безоговорочно доверять информации, полученной из сети. Это особенность можно использовать для распространения дезинформации за счет ее размещения на тех серверах, ссылки на которые являются для клиента подтверждением достоверности информации. Иногда создаются бесконечные циклы для автоматической загрузки других HTML- файлов. Несложная комбинация указателей в двух или более файлах позволяет создать замкнутый круг, когда браузер будет постоянно загружать файлы, не отображая при этом никакой полезной информации.

Большие возможности для несанкционированного доступа к Web- страницам предоставляет язык создания Web-приложений Java. Средства поддержки приложений, написанных на Java, называются апплетами (applets). Поскольку в Web-документе никак не сообщается, что за той или иной кнопкой скрывается Java-апплет, то пользователь заранее не может определить, что произойдет дальше при выборе необходимого документа или операции.

Существуют также методы несанкционированного доступа на WWW-серверы. Среди них можно отметить такие как переполнение буфера входных/выходных данных сервера, что приводит к повреждению данных или фрагментов кода, полное выведение Web-сервера из строя путем помещения в оперативную память сервера недопустимых команд, считывание файла паролей сервера, уничтожение журнала регистрации работы пользователей в сети и т.д.

3.17.3 Методы и средства защиты информации в компьютерных сетях

Накопленный опыт технологий защиты информации в компьютерных сетях показывает, что только комплексный подход к защите информации может обеспечить современные требования безопасности.

Комплексный подход подразумевает комплексное развитие всех методов и средств защиты.

Рассмотрим кратко основные методы и средства обеспечения безопасности информации в компьютерных сетях.

Методы защиты информации делятся:

- препятствия
- управление доступом
- маскировка
- регламентация
- принуждение
- побуждение.

Препятствие – метод физического преграждения пути злоумышленнику к защищаемой информации (компьютеру, сетевому оборудованию).

Управление доступом – метод защиты информации регулированием использования всех ресурсов системы. Управление доступом включает следующие функции защиты:

-идентификация пользователей, персонала и ресурсов системы, путем присвоения каждому объекту персонального идентификатора;

- опознавание объекта или субъекта по предъявляемому им идентификатору;

- проверка полномочий на запрашиваемые ресурсы;

- регистрация обращений к защищаемым ресурсам;

- реагирование при попытках несанкционированных действий.

Маскировка – метод защиты информации с помощью ее криптографического закрытия (шифрования). В настоящее время этот метод является наиболее надежным.

Известны три основных алгоритма: алгоритм DES, современный алгоритм Clipper (Capston) и так называемая общественная инициатива – алгоритм PGP.

Алгоритм шифрования DES (DataEncryptionStandard) был разработан в начале 70- х годов. Алгоритм шифрования был реализован в виде интегральной схемы с длиной ключа в 64 символа

(56 символов используются непосредственно для алгоритма шифрования и 8 для обнаружения ошибок).

Расчет алгоритмов в то время показывал, что ключ шифрования может иметь 72 квадриллиона комбинаций. Алгоритм DES был принят в США в качестве федерального стандарта обработки информации в 1977 году, а в середине 80-х был утвержден как международный стандарт, который каждые пять лет проходит процедуру подтверждения. Для оценки уровня защиты информации аналитики приводят такие факт: современный компьютер стоимостью 1 млн. долларов раскроет шифр за 7 часов, стоимостью 10 млн. долларов – за 20 минут, 100 млн. долларов – за 2 минуты. Агентство национальной безопасности США имеет такой компьютер.

Новый метод шифрования информации – технология Clipper– разработан агентством национальной безопасности США для защиты от прослушивания телефонных разговоров.

Для защиты данных этот метод носит название Capston. В основе метода положен принцип двух ключей-микросхем, обеспечивающих шифрование информации со скоростью до 1 гигабита в секунду. Пользователи получают ключи в двух пунктах, управляемых правительственными органами или частными концернами. Система ключей состоит из двух интегральных схем "Clipperchip" и "Capstonchip" и алгоритма шифрования SKIPJACK. Алгоритм шифрования шифрует символьные блоки данных с помощью 80-символьного ключа в 32 прохода. Он в 16 миллионов раз мощнее алгоритма DES и считается, только через несколько десятков лет компьютеры стоимостью 100 млн долларов смогут расшифровывать информацию за 2 минуты. Для сети Интранет разработан специальный протокол шифрования SKIP (SimpleKeymanagementforInternetProtocol), управляющий шифрованием потоков информации.

Отметим, что в настоящее время федеральные власти США запрещают экспорт протокола SKIP, поэтому во многих странах предпринимаются попытки создания его аналога.

Криптографические программные средства PGP (PrettyGoodPrivacy) были разработаны в 1991 году американским программистом Ф. Циммерманном для зашифровки сообщений электронной почты. Программа PGP свободна для доступа в Интернет и может быть установлена на любой компьютер. Принцип работы программы PGP основан на использовании двух программ-ключей:

одной у отправителя, а другой у получателя. Программы-ключи защищены не паролями, а шифровальной фразой. Расшифровать сообщение можно, только используя два ключа. Программа PGP использует сложный математический алгоритм, что вместе с принципом использования двух ключей делает дешифрацию практически невозможной. Появление программ PGP вызвало скандал в правоохранительных кругах США, так они лишают возможности контроля за информацией.

Отметим, что криптографические алгоритмы широко используются для защиты электронной цифровой подписи.

Более полную информацию о криптографических методах можно получить на сайте www.cripto.com или www.confident.ru.

Регламентация – метод защиты информации, создающий такие условия автоматизированной обработки, хранения и передачи защищаемой информации, при которых возможности несанкционированногодоступа к ней сводился бы к минимуму.

Принуждение – такой метод защиты информации, при котором пользователи и администраторы сети вынуждены соблюдать правила обработки, передачи и использования защищаемой информации подугрозой материальной, административной или уголовной ответственности.

Побуждение – метод защиты, который побуждает пользователей и администраторов сети не нарушать установленных за счет соблюдения моральных и этических норм.

Средства защиты информации делятся:

- технические средства
- программные средства
- организационные средства
- морально- этические
- законодательные.

3.17.4 Защита информации в сети Интернет

Несмотря на предпринимаемые меры, проблема несанкционированного доступа к ресурсам Интернет до конца не решена, хотя и сформулирован ряд положений по обеспечению безопасности обработки информации.

Авторизация доступа к Интернет предназначена прежде всего для учета использования ее ресурсов и оплаты услуг. Поэтому, как правило, авторизация доступа осуществляется провайдерами и предназначена исключительно для коммерческого использования.

Различные провайдеры предоставляют своим клиентам различную степень свободы. Так, например, при выделенном TCP/IP подключении пользователь оказывается привязанным к конкретному географическому адресу.

При подключении по коммутируемым каналам связи выход в сети может быть осуществлен с любого телефонного аппарата. Кроме этого конкуренция провайдеров на рынке привела к новому виду предоставления услуг – гостевому, позволяющему работать в сети без регистрации и оплаты. В этом случае сетевой адрес назначается провайдером динамически и проконтролировать такого пользователя, если он занимается хакерской деятельностью, является достаточно сложной задачей. Единственной возможностью является анализ регулярных попыток пользователя получить доступ к ресурсам, которые не пользуются популярностью у других абонентов.

Основным средством защиты сети Интернет от несанкционированного доступа в настоящее время является *файрволл*. Как отмечалось в предыдущих лекциях, файрволлы контролируют информационные потоки между локальными сетями, причем уровень контроля определяется в первую очередь сферой интересов компании, структурой локальной сети и целями, ради которых локальная сеть подключается к Интернет.

Система файрволл обеспечивает защиту программного обеспечения сервера и пользователей от доступа без соответствующей авторизации, но в то же время не препятствует нормальной работе штатных протоколов (электронная почта, FTP, WWW и др.).

Во многих организациях файрволл используется для хранения данных с открытым доступом, например, информация о продуктах и услугах, сообщения об ошибках и т.д. Дополнительный контур защиты, организованный с помощью файрволла, реализуется за счет объединения точек контроля доступа и обеспечения безопасности в одном и том же месте, как аппаратно, так и программно.

Отметим, что системы класса файрволл не в состоянии обеспечить защиту от вирусов и червей, они также беззащитны перед атакой, когда в поражаемую систему под видом сетевой почты копируется какая-либо программа, которая после открытия почтового ящика запускается на выполнение.

Существует метод защиты локальной сети, который используется для защиты сетей крупных банков в США при их

подключении кИнтернет, когдаблокируются все внешние потоки информации и в локальную сеть поступают лишь копии разрешенных сообщений.

В этом случае между локальной сетью и сетью Интернет нет точек непосредственного соприкосновения, а весь поток информации обрабатывается proxi- сервером. В случае, если с любого компьютера локальной сети поступает запрос к ресурсам Интернет, то proxi – сервер проверяет у пользователя права доступа к Интернет. В ходе сеанса связи система безопасности регистрирует всех пользователей внешней связью, имена пересылаемых файлов, все сообщения копируются. Предусмотрено сохранение более подробной информации.

Как правило, для ограничения доступа внешних пользователей к локальной сети составляются таблицы контроля доступа, так называемыйaccesscontrollists, которые используются маршрутизаторами при получении пакетов. Отметим, что создавать такие таблицы довольно сложно, а их просмотр снижает производительность сети.

По оценке специалистов системы безопасности, установленные на маршрутизаторах, менее надежны, чем основанные на применении файрволлов и proxy-серверов. С другой стороны, сами файрволлы и proxi- сервера могут быть атакованы взломщиками. Поэтому оптимальным является комбинированный подход: маршрутизатор отвергает нежелательные запросы доступа и передает все остальные запросы файрволлу и далее proxi-серверу.

В перспективе файрволлы должны уступить место встроенным системам безопасности, протоколы Интернет должны предусмотреть процедуры, обеспечивающие проверку прав пользователей, целостности сообщений, а также шифрование информации. В настоящее время комитет по стандартам Интернет работают над такими протоколами, но совершенно не ясно, когда закончится эта работа. Кроме этого ведутся работы по новому программному обеспечению маршрутизаторов, где будет использоваться метод обмена ключами.

В целом можно констатировать, что в ближайшем будущем ожидается интенсивное развитие новых направлений в методах и технологиях защиты информации в сети Интернет, в основе которых лежит принцип не разрозненных решений, а концепция интегральной безопасности.

3.17.5Стандарты безопасности информации

Основы стандартов на безопасность были заложены изданными в 1983 году "Критериями оценки надежных компьютерных систем". Этот документ, изданный в США национальным центром компьютерной безопасности (NCSC – NationalComputerSecurityCenter), часто называют Оранжевой Книгой. Утвержденная в 1985 году в качестве правительственного стандарта, Оранжевая Книга определяет основные требования и специфицирует классы для оценки уровня безопасности готовых и коммерчески поддерживаемых компьютерных систем.

В соответствии с требованиями Оранжевой книги, безопасной считается такая система, которая "посредством специальных механизмов защиты контролирует доступ к информации таким образом, что только имеющие соответствующие полномочия лица или процессы, выполняющиеся от их имени, могут получить доступ на чтение, запись, создание или удаление информации".

Иерархия надежных систем, приведенная в Оранжевой Книге, помечает низший уровень безопасности как С, высший как А, промежуточный как В. В класс D попадают системы, оценка которых выявила их несоответствие требованиям всех других классов.

Основные свойствами, характерными для С-систем, являются: наличие подсистемы учета событий, связанных с безопасностью, избирательный контроль доступа. Избирательный контроль заключается в том, что каждый пользователь в отдельности наделяется или лишается привилегий доступа к ресурсам. Уровень С делится на 2 подуровня: С1 и С2. Уровень С2 предусматривает более строгую защиту, чем С1. В соответствии с этим уровнем требуется отслеживание событий, связанных с нарушениями защиты, детальное определение прав и видов доступа к данным, предотвращение случайной доступности данных (очистка освобожденной памяти). На уровне С2 должны присутствовать средства секретного входа, которые позволяют пользователям идентифицировать себя путем ввода уникального идентификатора (ID) входа и пароля перед тем, как им будет разрешен доступ к системе.

Требования уровней В и А гораздо строже и редко предъявляются к массово используемым продуктам.

Различные коммерческие структуры (например, банки) особо выделяют необходимость аудита, службы, соответствующей рекомендации С2. Любая деятельность, связанная с безопасностью,

может быть отслежена и тем самым учтена. Это как раз то, что требует С2, и то, что обычно нужно банкам. Однако, коммерческие пользователи, как правило, не хотят расплачиваться производительностью за повышенный уровень безопасности.

Уровень безопасности А занимает своими управляющими механизмами до 90% времени компьютера. Более безопасные системы не только снижают эффективность, но и существенно ограничивают число доступных прикладных пакетов, которые соответствующим образом могут выполняться в подобной системе.

Вопросы для самоконтроля

1. Дайте определение термину «Интернет».
2. Выделите четыре основных аспекта в историческом развитии сети Интернет.
3. Кто стоял у истоков создания сети Интернет?
4. Назовите первое приложение для сети ARPANET– прародительницы сети Интернет.
5. Какие четыре принципа легли в основание протоколаTCP/IP?
6. Какие преимущества дает функционированию глобальной сети принцип «на локальном уровне не должно существовать глобальной системы управления»?
7. В чем суть ключевой концепции создания Интернет?
8. Какую роль в информационной революции приняли и еще примут

Web-серверы и облачные технологии?

9. Чем занимается организация IAB?
10. Какую функцию выполняет подразделение Интернета InterNIC?
11. Что такое W3-консорциум и каковы его обязанности?
12. Опишите соответствие уровней модели OSIи уровней протокола TCP/IP.
13. За счет чего протокол TCP обеспечивает надежность доставки сообщений и составляющих их пакетов? Что такое квитанция?
14. Что такое время тайм-аута и «размер скользящего окна» в протоколе TCP?
15. Зачем, на ваш взгляд, в протоколе TCP ввели типологизацию прикладных программ с помощьюпонятия «порта»?
16. Что определяет понятие «сокет»?
17. Какими двумя видами очередей пакетов управляет протокол TCP?

18. Что характеризует IP-адрес и может ли маршрутизатор иметь больше одного IP-адреса?
19. Перечислите выделенные IP-адресаи их предназначения.
20. С какой целью используются «серые» адреса?
21. В связи с чем пришлось ввести протокол IPv6?
22. Почему сети класса С являются наиболеераспространенными сетями?
23. Почему сети класса D называются сетями multicast?
24. Почему использование классов IP-адресов напрямую нецелесообразно? Зачем понадобился инструмент масок? Что такое маска?
25. Пусть IP-адрес сети класса В задан в виде 135.56.217.7. Используется маска 255.255.192.0. Определите новый номер сети и номер узла в ней.
26. Пусть в сети работают два компьютера с IP-адресами 210.20.30.193 и 210.20.30.70. Для разделения этих компьютеров в две разные подсети используем маску 255.255.255.240. Определите новые адреса подсетей и соответствующие адреса узлов в этих подсетях.
27. Что такое правило маршрутизации?
28. Какие три разных элемента могут входить в состав столбца таблицы маршрутизации «сетевой адрес»?
29. С помощью команд routeprint и ipconfig (в командной строке) проанализируйте таблицу маршрутизации на вашем домашнем компьютере без подключения к сети Интернет и после подключения к ней. Опишите назначения разных сетевых адресов. Почему адреса шлюзов и интерфейсов иногда могут не совпадать?
30. Каково назначение в сети компьютера WinProxy?
31. Что такое домен?
32. Опишите структуру базы данных в системе DNS?
33. Какие механизмы передачи данных поддерживает протокол SMTP?
34. Чем отличается работа почтовых протоколов POP3 и IMAP4?
35. Каково предназначение сервиса URL при работе с WWW-сервером?
36. Что такое HTML? Что собой представляет HTML-файл? В чем суть основной концепции размещения информации на Web-странице?
37. Дайте определение Web-сайта.
38. Для чего используется сервисы IDC и ASP?

39. Опишите принцип работы HTTP-протокола.
40. За счет чего промежуточный агент Proxy ускоряет работу Интернет?
41. В каком случае протокол HTTP использует объект Tunnel?
42. Какие два канала соединения поддерживает протокол FTP?
43. Что позволяет делать протокол TELNET с удаленным компьютером?
44. На каком уровне (прикладном, сетевом или канальном) чаще всего работают файрволлы и почему?
45. Благодаря чему сотни крупных провайдеров связывают свои магистральные сети в единое информационное пространство?
46. Перечислите различные приемы нарушения безопасности сети.
47. Чем отличаются друг от друга программы «Троянский конь», «Вирус», «Червь», «Жадная»?
48. Чем отличаются конструктивный и деструктивный несанкционированный доступы?
49. За счет чего, как правило, взламывается электронная почта?
50. Почему Java-апплеты обладают большими возможностями несанкционированного доступа к Web-страницам.
51. Почему криптографическая программа PGP делает дешифрацию почти невозможной?
52. Почему системы класса файрлволл не в состоянии обеспечить защиту от вирусов, червей и посторонних программ?
53. Что придумали крупные банки США для защиты при работе в Интернете?
54. В чем, на ваш взгляд, видятся плюсы и минусы распространения облачных технологий в плане информационной безопасности?

Список использованных источников

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Ученик для вузов. 3-е изд. ./ СПб.: Питер, 2006. – 958с.

2. Олифер В.Г., Олифер Н.А. Сетевые операционные системы. Учебник./ СПб.: Питер, 2006. – 544с.

3. Таненбаум Э. Компьютерные сети. / М. «Вильямс», 2003. – 992с.

4. Вишневский В.М. Теоретические основы проектирования компьютерных сетей. / Москва: Техносфера, 2003. – 512с.

5. Уэнделл О. Компьютерные сети. Первый шаг. / Москва: Издательский дом «Вильямс», 2006. – 432с.

6. Сосновский О.А. Компьютерные сети и сетевые технологии. Курс лекций. / БГЭУ.: Минск, 2003. – 133с.

Оглавление

КОМПЬЮТЕРНЫЕ СЕТИ

Курс лекций
по одноименной дисциплине
для слушателей специальностей 1-40 01 74
«Web-дизайн и компьютерная графика» и 1-40 01 73
«Программное обеспечение информационных систем»
заочной формы обучения

Составитель **Осипенко** Александр Николаевич

Подписано к размещению в электронную библиотеку
ГГТУ им. П. О. Сухого в качестве электронного
учебно-методического документа 18.02.15.

Рег. № 156Е.
http://www.gstu.by