

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный технический университет
имени К.И. Сатпаева

Г.Д. Жангисина, Х.И. Юбузова

КОМПЬЮТЕРНЫЕ СЕТИ

Учебное пособие
для специальностей «Вычислительная техника и программное
обеспечение», «Системы информационной безопасности»

Рекомендовано Научно-методическим советом
университета в качестве учебного пособия

Алматы 2014

УДК 004 (075.8)
ББК 32.973я73

Ж27 Жангисина Г.Д., Юбузова Х.И. Компьютерные сети. Учеб. пособие. – Алматы: КазНТУ, 2014. – 182 с.

Ил. 42. Табл. 4. Библиогр. – 20 назв.

ISBN 978-601-228-629-8

В учебном пособии рассматриваются основные аспекты архитектуры и технологии современных компьютерных сетей. В доступной форме излагаются теоретические положения и принципы построения компьютерных сетей, особенности традиционных и перспективных технологий проводных, беспроводных локальных сетей, способы создания составной сети, глобальной сети, популярных сетевых услуг и сервисов, вопросы организации безопасности.

Учебное пособие предназначено для студентов специальностей 5В100200 «Системы информационной безопасности» и 5В070400 «Вычислительная техника и программное обеспечение». Пособие может быть полезным для аспирантов, преподавателей вузов, а также для студентов других специальностей и технических специалистов.

Рецензенты:

Пак И. Т., г.н.с. Института проблем информатики и управления МОН РК, д-р техн. наук, профессор

Имангалиев Ш. И., зав. кафедрой «Информационные системы», канд. техн. наук, доцент

Алматинского университета энергетике и связи

Казиев Г. З., директор РИПК, д-р техн. наук, профессор
КазНТУ им. К. И. Сатпаева

Печатается по плану издания Министерства образования и науки Республики Казахстан на 2014 г.

ISBN 978-601-228-629-8

© Жангисина Г.Д., Юбузова Х.И., 2014

© КазНТУ, 2014

ВВЕДЕНИЕ

Компьютерные сети в современном обществе являются самыми востребованными ресурсами. Войдя в человеческую жизнь, компьютеры сейчас стали неотъемлемой частью нашей цивилизации.

С развитием компьютерной техники расширяется и сфера ее использования, изменяется и терминология. Термин компьютерная система в литературе имеет различные определения: от просто набора устройств обработки данных (автоматизированных или автоматических), от одиночного компьютера с его программным обеспечением, до совокупности нескольких взаимосвязанных компьютеров с их программным обеспечением и периферийным оборудованием, предназначенным для сбора, хранения, обработки и распределения информации.

Начиная с 70-х годов, компьютеры оснащаются многочисленными внешними устройствами, которые в совокупности составляют систему.

Компьютерная система – совокупность одного и более компьютеров или процессоров, программного обеспечения и периферийного оборудования, организованная для совместного выполнения информационно-вычислительных процессов.

Компьютерные сети привнесли в телекоммуникационный мир неисчерпаемые запасы информации, созданные цивилизацией за несколько тысячелетий своего существования и продолжающие пополняться с растущей скоростью в наши дни.

Компьютерные сети являются привычным инструментом коммуникаций, информационного обмена и выполнения вычислений. Одновременно организация сетей, принципы их функционирования и правила конфигурирования не являются простыми и требуют от специалистов, выполняющих проектирование, построение и администрирование компьютерных

сетей, определенных теоретических знаний и обязательного практического опыта.

Материал учебного пособия охватывает общие понятия о компьютерных сетях, их структуре, сетевых компонентах. Здесь приводятся схемы основных видов топологий, используемых для физического соединения компьютеров в сети, методы доступа к каналу связи, физические среды передачи данных. Передача данных в сети рассматривается на базе эталонной базовой модели, разработанной Международной организацией по стандартам взаимодействия открытых сетей. Описываются правила и процедуры передачи данных между информационными системами. Приводятся типы сетевого оборудования, их назначение и принципы работы. Описывается сетевое программное обеспечение, используемое для организации сетей. Рассматриваются принципы межсетевого взаимодействия. Приводятся основные понятия из области сетевой безопасности.

Для подготовки курса проработан большой объем информации, расположенной на информационно-поисковых серверах Internet и использовалась литература, приведенная в списке.

В первой главе рассматриваются вопросы, связанные с эволюцией компьютерных сетей, понятие глобальной и локальной сетей, их сближение и конвергенция. За основу были взяты материалы из [1, 13, 15].

Во второй главе объясняются понятия топологии, приводятся виды топологий, их достоинства и недостатки, коммутация и маршрутизация, рассматриваются проблемы объединения узлов в сеть и их адресация в сети. Основные источники [1, 7, 13].

Третья глава рассматривает проблемы связанные с принципами коммутации пакетов, каналов, рассматриваются вопросы организации виртуального канала. Основные источники [1, 7, 8, 12, 13].

В четвертой и пятой главах рассматривается архитектура сетей и ее взаимодействие с семиуровневой базовой эталонной моделью связи открытых систем (OSI). Представлен каждый уровень, его функции и протоколы, используемые на каждом уровне. Описаны методы доступа к каналу связи и их использование. Стандарты технологии Ethernet. При подготовке были взяты материалы [5, 7, 8, 12, 13, 14].

В шестой главе описаны беспроводные сети и спецификации стандартов IEEE802. Основные источники [6, 7, 9, 10, 11, 15, 17].

В седьмой, восьмой и девятой главах даются основные сведения по построению интерсети, применению протоколов стека ТСР/ІР, порядке назначения и разрешения адресов, протоколе межсетевого взаимодействия. При подготовке были взяты материалы [1, 7, 12, 13].

Глава десятая и одиннадцатая посвящена вопросам, связанным с сетевыми службами, технологиями и протоколами интернета, а также безопасности ІР сети. Основные источники [1, 7, 8, 12, 18].

1. ЭВОЛЮЦИЯ КОМПЬЮТЕРНЫХ СЕТЕЙ

Компьютерные сети, называемые также информационными сетями, или сетями передачи данных, являются логическим результатом эволюции двух важнейших научно-технических отраслей современной цивилизации – компьютерных и телекоммуникационных технологий. С одной стороны, сети представляют собой частный случай распределенных вычислительных систем, в которых группа компьютеров согласованно выполняет набор взаимосвязанных задач, обмениваясь данными в автоматическом режиме. С другой стороны, компьютерные сети могут рассматриваться как средства передачи информации на большие расстояния, для чего в них применяются методы кодирования и мультиплексирования данных, получившие развитие в различных телекоммуникационных системах.

1.1. Первые глобальные сети

Потребность в соединении компьютеров, находящихся на расстоянии друг от друга, начиналась с решения более простой задачи – доступа к компьютеру с терминалов на многие сотни, а то и тысячи километров. Терминалы соединялись с компьютерами через телефонные сети с помощью модемов. Затем появились системы, в которых наряду с удаленными соединениями типа терминал-компьютер реализованы, и удаленные связи типа компьютер-компьютер, получившие возможность обмениваться данными в автоматическом режиме, что, и является базовым механизмом любой вычислительной сети. На основе этого механизма в первых сетях были реализованы службы обмена файлами, синхронизации баз данных, электронной почты и др., ставшие теперь традиционными сетевыми службами.

Таким образом, хронологически первыми появились глобальные сети, т. е. сети, объединяющие территориально рассредоточенные компьютеры, возможно находящиеся в различных городах и странах. С построения глобальных сетей были впервые предложены и отработаны основные идеи и концепции современных вычислительных сетей (многоуровневое построение коммуникационных протоколов, концепции коммутации и маршрутизации пакетов).

Глобальные компьютерные сети очень многое унаследовали от более старых и распространенных глобальных сетей – телефонных. Результатом создания первых глобальных компьютерных сетей стал отказ от принципов коммутации каналов, на протяжении многих десятков лет применяемого в телефонных сетях.

Выделяемый на все время сеанса связи составной канал не мог эффективно использоваться пульсирующим трафиком компьютерных данных, у которого периоды интенсивного обмена чередуются с продолжительными паузами. Натурные эксперименты и математические модели показали, что пульсирующий компьютерный трафик гораздо эффективнее передается сетями, работающих с помощью принципа коммутации пакетов, когда данные разделяются на небольшие порции – пакеты.

В течение многих лет глобальные сети строились на основе телефонных каналов тональной частоты, способных в каждый момент времени вести передачу только одного разговора в аналоговой форме. А так как скорость передачи дискретных компьютерных данных по таким каналам была очень низкой (десятки килобит в секунду), набор предоставляемых услуг в глобальных сетях такого типа обычно ограничивался передачей файлов, преимущественно в фоновом режиме, и электронной почтой. Помимо низкой скорости такие каналы имеют и другой недостаток – они вносят значительные искажения в передаваемые сигналы. Поэтому протоколы глобальных сетей, построенные с использованием каналов связи низкого качества, отличаются сложными процедурами контроля и восстановления данных. Типичный пример таких сетей – сети X.25, разработанные в начале 70-х годов. Эти аналоговые каналы были низкоскоростными и были преобладающим типом каналов, соединяли компьютеры и коммутаторы глобальной вычислительной сети.

Прогресс глобальных компьютерных сетей во многом определялся прогрессом телефонных сетей. С конца 60-х годов в телефонных сетях часто стала применяться передача голоса в цифровой форме, в дальнейшем это привело к появлению высокоскоростных цифровых каналов, соединяющих АТС и позволяющих одновременно передавать десятки и сотни разговоров. Была разработана специальная технология плезиохронной цифровой иерархии (РОН), предназначенная для создания так называемых первичных, или опорных, сетей. Такие сети не предоставляют услуг

конечным пользователям, они являются фундаментом, на котором строятся скоростные цифровые каналы «точка-тире-точка», соединяющие оборудование другой (так называемой наложенной) сети, уже работающая на конечного пользователя.

Первоначально технология РОН, поддерживающая скорости до 140 Мбит/с, была внутренней технологией телефонных компаний. Однако со временем эти компании стали сдавать часть своих каналов РОН в аренду предприятиям, применяемых ими для создания собственных телефонных и глобальных компьютерных сетей.

Появившаяся в конце 80-х годов технология синхронной цифровой иерархии расширила диапазон скоростей цифровых каналов до 10 Гбит/с, а технология спектрального мультиплексирования – до сотен гигабит и даже нескольких терабит в секунду.

В настоящее время глобальные сети по разнообразию и качеству сервисов обогнали локальные сети, которые долгое время были лидерами в этом отношении, хотя и появились на свет значительно позже.

1.2. Первые локальные сети

В начале 70-х годов, в результате технологического прорыва в области производства компьютерных компонентов появились большие интегральные схемы (БИС). Их сравнительно невысокая стоимость и хорошие функциональные возможности привели к созданию мини-компьютеров, которые стали реальными конкурентами мэйнфреймов.

Даже небольшие подразделения предприятий получили возможность иметь собственные компьютеры. Мини-компьютеры решали задачи управления технологическим оборудованием, складом и другие задачи предприятия. Таким образом, появилась концепция распределения компьютерных ресурсов по всему предприятию. Однако при этом все компьютеры одной организации по-прежнему продолжали работать *автономно*.

Со временем росли и потребности пользователей в работе с вычислительной техникой. Возникла необходимость режима автоматического обмена компьютерными данными с пользователями других подразделений. Ответом на эту потребность стало появление первых локальных сетей.

Локальные сети (Local Area Network, LAN) – объединение компьютеров, сосредоточенных на небольшой территории, обычно в радиусе не более 1–2 км, хотя в отдельных случаях локальная сеть может иметь и более протяженные размеры, например, несколько десятков километров. В общем случае локальная сеть представляет собой коммуникационную систему, принадлежащую одной организации.

На первых порах для соединения компьютеров друг с другом использовались нестандартные сетевые технологии.

Сетевая технология – это согласованный набор программных и аппаратных средств (например, драйверов, сетевых адаптеров, кабелей и разъемов) и механизмов передачи данных по линиям связи, достаточный для построения вычислительной сети.

Разнообразные устройства сопряжения, использующие собственные способы представления данных на линиях связи, свои типы кабелей и тому подобные могли соединять только те конкретные модели компьютеров, для которых были разработаны, например, миникомпьютеры PDP-11 с мэйнфреймом IBM 360.

В середине 80-х годов положение дел в локальных сетях кардинально изменилось. Утвердились стандартные сетевые технологии объединения компьютеров в сеть – Ethernet, ArcNet, Token Ring, Token Bus, несколько позже – FDDI.

Все стандартные технологии локальных сетей опирались на тот же принцип коммутации, который был с успехом опробован и доказал свои преимущества при передаче трафика данных в глобальных компьютерных сетях – принцип коммутации пакетов.

Стандартные сетевые технологии значительно упростили процесс построения локальной сети. Для создания сети достаточно было приобрести стандартный кабель, сетевые адаптеры соответствующего стандарта, например Ethernet, вставить адаптеры в компьютеры, присоединить их к кабелю стандартными разъемами.

Последствием и одновременно движущей силой прогресса локальных сетей стало появление огромного числа непрофессиональных пользователей, освобожденных от необходимости изучать специальные и сложные команды для сетевой работы.

Главным образом, это связано с использованием в локальных сетях качественных кабельных линий связи, на которых даже

сетевые адаптеры первого поколения обеспечивали скорость передачи данных до 10 Мбит/с.

Конец 90-х годов выявил явного лидера среди технологий локальных сетей – семейство Ethernet, в которое вошли классическая технология Ethernet 10 Мбит/с, а также Fast Ethernet 100 Мбит/с и Gigabit Ethernet 1000 Мбит/с.

Простые алгоритмы работы предопределили низкую стоимость оборудования Ethernet. Широкий диапазон иерархии скоростей позволил рационально строить локальную сеть, выбирая ту технологию семейства, которая в наибольшей степени отвечала задачам предприятия и потребностям пользователей. Важно также, что все технологии Ethernet очень близки друг к другу по принципам работы, что упрощает обслуживание и интеграцию этих сетей.

Хронологическую последовательность важнейших событий, ставших историческими вехами на пути появления первых компьютерных сетей, иллюстрирует табл. 1.1.

Таблица 1.1.

Хронология появления первых компьютерных сетей

Этап	Время
Первые глобальные связи компьютеров, первые эксперименты с пакетными сетями	Конец 60-х годов
Начало передач по телефонным сетям голоса в цифровой форме	Конец 60-х годов
Появление больших интегральных схем, первые мини-компьютеры. Первые нестандартные локальные сети	Начало 70-х годов
Стандартизация технологии X.25	1974 г.
Появление персональных компьютеров, создание Интернета в современном виде, установка на всех узлах стека TCP/IP	Начало 80-х годов
Появление стандартных технологий локальных сетей (Ethernet – 1984 г., Token Ring – 1985 г., FDDI – 1989 г.)	Середина 80-х годов
Начало коммерческого использования Интернета	Конец 80-х годов
Изобретение Web	1991 г.

1.3. Сближение локальных и глобальных сетей

В конце 80-х годов отличия между локальными и глобальными сетями проявлялись весьма отчетливо.

Протяженность и качество линий связи. Локальные компьютерные сети по определению отличаются от глобальных сетей небольшими расстояниями между узлами сети. Это в принципе делает возможным использование в локальных сетях более качественных линий связи.

Сложность методов передачи данных. В условиях низкой надежности физических каналов в глобальных сетях требуются более сложные, чем в локальных сетях, методы передачи данных и соответствующее оборудование.

Скорость обмена данными в локальных сетях в то время была существенно выше, чем в глобальных.

Разнообразие услуг. Широкий набор услуг – это различные виды услуг файловой службы, услуги печати, услуги баз данных, электронная почта и др., глобальные сети в основном предоставляли почтовые услуги и иногда файловые услуги с ограниченными возможностями.

Масштабируемость. «Классические» локальные сети обладали плохой масштабируемостью из-за жесткости базовых топологий, определяющих подключения станций и длину линии. При этом характеристики сети ухудшаются при достижении определенного предела по количеству узлов или протяженности линий связи. Глобальным сетям присуща хорошая масштабируемость, т. к. они изначально разрабатывались с произвольными топологиями и сколь угодно большим количеством абонентов.

Постепенно различия между локальными и глобальными типами технологий стали сглаживаться. Изолированные ранее локальные сети начали объединять друг с другом, при этом глобальные сети используют в качестве связующей среды. Тесная интеграция локальных и глобальных сетей способствовала значительному взаимопроникновению соответствующих технологий.

Сближение в методах передачи данных происходит на платформе модулированной передачи данных по волоконно-

оптическим кабелям. Эту среду передачи данных используют практически все технологии сетей для скоростного обмена информацией на расстояниях свыше 100 м, на ней же построены современные магистрали первичных сетей SDN и DWDM, предоставляющих свои цифровые каналы для объединения оборудования глобальных компьютерных сетей.

В локальных сетях в последнее время уделяется такое же большое внимание методам обеспечения защиты информации от несанкционированного доступа, как и в глобальных сетях. Это обусловлено тем, что локальные сети перестали быть изолированными, они имеют выход в «большой мир» через глобальные связи. Защита локальных сетей часто строится на методах – шифрование данных, аутентификация и авторизация пользователей. И, наконец, появляются новые технологии, изначально предназначенные для обоих видов сетей. Ярким представителем нового поколения технологий явилась технология ATM, которая может служить основой для глобальных и локальных сетей, эффективно объединять все существующие типы трафика в одной транспортной сети. Другим примером может служить семейство технологий Ethernet, имеющее явные «локальные» корни. Новый стандарт Ethernet 10G, позволяющий передавать данные со скоростью 10 Гбит/с, предназначен для магистралей как глобальных, так и крупных локальных сетей.

Одним из проявлений сближения локальных и глобальных сетей является появление сетей масштаба большого города, занимающих промежуточное положение между локальными и глобальными сетями. *Городские сети*, или *сети мегаполисов*, предназначены для обслуживания территории крупного города. Эти сети используют цифровые линии связи, часто оптоволоконные, со скоростями на магистрали от 155 Мбит/с и выше, обеспечивают экономичное соединение локальных сетей между собой, а также выход в глобальные сети. Первоначально разработанные для передачи данных, сейчас они поддерживают такие услуги, как видеоконференции, интегральная передача голоса и текста. Для сетей мегаполисов даже был разработан специальный протокол – SMDS, но позднее он был вытеснен более мощной технологией ATM. Современные сети типа MAN

отличаются разнообразием предоставляемых услуг, позволяя своим клиентам объединять коммуникационное оборудование различного типа, в т. ч. и офисные АТС.

1.4. Конвергенция компьютерных и телекоммуникационных сетей

К телекоммуникационным сетям, кроме компьютерных, относятся телефонные сети, радиосети и телевизионные сети. Во всех них в качестве ресурса, предоставляемого клиентам, выступает информация.

Конвергенция телекоммуникационных сетей идет по многим направлениям. Прежде всего, наблюдается *сближение видов услуг*.

Компьютерные сети изначально разрабатывались для передачи цифровой информации, которую часто называют просто данными.

Телефонные сети оказывают интерактивные услуги, т. к. два абонента, участвующие в разговоре (или несколько абонентов), попеременно проявляют активность.

Радиосети и *телевизионные сети* оказывают широкоэмитательные услуги, при этом информация распространяется только в одну сторону – из сети к абонентам, по схеме «один ко многим».

Телефонные сети и радиосети созданы для передачи голосовой информации, а телевизионные сети передают голос и изображение.

Первая попытка создания универсальной, так называемой *мультисети*, способной оказывать различные услуги, привела к появлению технологии цифровых сетей с интегральными услугами – ISDN. Сегодня на роль глобальной мультисервисной сети нового поколения претендует Интернет.

Технологическое сближение сетей происходит сегодня на основе цифровой передачи информации различного типа, метода коммутации пакетов и программирования услуг.

Сегодня становится все более очевидным, что мультисервисная сеть нового поколения не может быть создана в результате победы какой-либо одной технологии или подхода.

Ее может породить только процесс конвергенции, а сплав и даст требуемое качество для поддержки существующих и создания новых услуг. Появился новый термин – *инфокоммуникационная сеть*, т. е. система информационной (компьютерной) сети и телекоммуникационной.

Выводы:

Компьютерные сети – логический результат эволюции компьютерных и телекоммуникационных технологий. С одной стороны, они являются частным случаем распределенных компьютерных систем, а с другой – представляют средства передачи информации на большие расстояния, применяют методы кодирования и мультиплексирования данных, получившие развитие в различных телекоммуникационных системах.

Сети можно классифицировать по территориальному признаку: глобальные (WAN), локальные (LAN) и городские (MAN).

Хронологически первыми появились сети WAN. Объединяли компьютеры, рассредоточенные на расстоянии сотен и тысяч километров. Первые глобальные компьютерные сети очень многое унаследовали от телефонных сетей, обладали низкими скоростями передачи данных и ограничивали набор предоставляемых услуг: передача файлов в фоновом режиме и электронная почта.

Сети LAN ограничены расстояниями несколько километров; строятся с использованием высококачественных линий связи, позволяющие применять более простые методы передачи данных, достигать высоких скоростей обмена данными – до нескольких гигабитов в секунду.

Сети MAN предназначены для обслуживания территории крупного города, обладают качественными линиями связи и поддерживают высокие скорости.

Важнейший этап в развитии сетей – появление стандартных сетевых технологий: Ethernet, FDDI, Token Ring, позволяющих быстро и эффективно объединять компьютеры различных типов.

В дальнейшем в результате тесной интеграции LAN, WAN и MAN произошло взаимопроникновение соответствующих технологий.

Контрольные вопросы:

1. История развития компьютерных сетей.
2. Объясните понятие «сетевая технология».
3. Какой формы данные передавались в глобальной сети, построенной на основе телефонных линий?
4. Какой недостаток имели каналы тональной частоты?
5. Что представляет собой технология плезиохронной цифровой иерархии?
6. Какой была первоначальная скорость технологии РОН?
7. В чем отличие технологии синхронной цифровой иерархии?
8. Какой диапазон скоростей обеспечивала технология синхронной цифровой иерархии?
9. В чем отличие локальных и глобальных сетей?
10. В чем проявляется сближение локальных и глобальных сетей?
11. Какую скорость передачи данных обеспечивали сетевые адаптеры первого поколения?
12. Какова причина плохой масштабируемости «классической» локальной сети?
13. Какова причина хорошей масштабируемости глобальной сети?
14. Какую среду передачи данных используют технологии сетей для скоростного обмена информацией на расстояния свыше 100 м?

2. ОБЩИЕ ПРИНЦИПЫ ПОСТРОЕНИЯ СЕТЕЙ

2.1. Основные определения

Среда передачи данных – совокупность линий передачи данных и блоков взаимодействия (сетевого оборудования, не входящего в станции данных), предназначенных для передачи данных между станциями. Среды передачи данных могут быть общего пользования или выделенными для конкретного пользователя.

Линии передачи данных – средства, которые используются в информационных сетях для распространения сигналов в нужном направлении (рис. 2.1). Примерами линий передачи данных являются коаксиальный кабель, витая пара проводов, световод и т. д. Характеристиками линий передачи данных являются зависимость затухания сигнала от частоты и расстояния. Затухание принято оценивать в децибелах, $1дБ=10*\lg(P_1/P_2)$, где P_1 и P_2 – мощности сигнала на выходе линии соответственно.

При заданной длине можно говорить о *полосе пропускания* (полосе частот) линии. Полоса пропускания связана со *скоростью передач информации*. Различают *бодовую (модуляционную)* и *информационную* скорости. Бодовая скорость измеряется в бодах, т. е. числом изменений дискретного сигнала в единицу времени, а информационная – числом битов информации, переданных в единицу времени. Именно бодовая скорость определяется полосой пропускания линии. Если на бодовом интервале (между соседними изменениями сигнала) передается N бит, то число градаций модулируемого параметра несущей равно 2^N . Например, при числе градаций 16 и скорости 1200 бод одному боду соответствует 4 бит/с и информационная скорость равна 4800 бит/с.

Канал (канал связи) – средства односторонней передачи данных. Примером канала, может быть, полоса частот, выделенная одному передатчику при радиосвязи. В некоторой линии можно образовать несколько каналов связи, по каждому из которых передается своя информация. При этом говорят, что линия разделяется между несколькими каналами.

Два метода разделения линии передачи данных: временное мультиплексирование (*Time Division Multiplexing, TDM*), при котором каждому каналу выделяется некоторый квант времени, и частотное разделение (*FDM, Frequency Division Method*), при котором каналу выделяется некоторая полоса частот.

Канал передачи данных – средства двустороннего обмена данными, включающие аппаратуру передачи данных (АПД) и линию передачи данных. По природе физической среды передачи данных (ПД) различают каналы передачи данных на оптических линиях связи, проводных (медных) линиях связи и беспроводные. В свою очередь, медные каналы могут быть представлены коаксиальными кабелями и витыми парами, а беспроводные радио- и инфракрасными каналами.

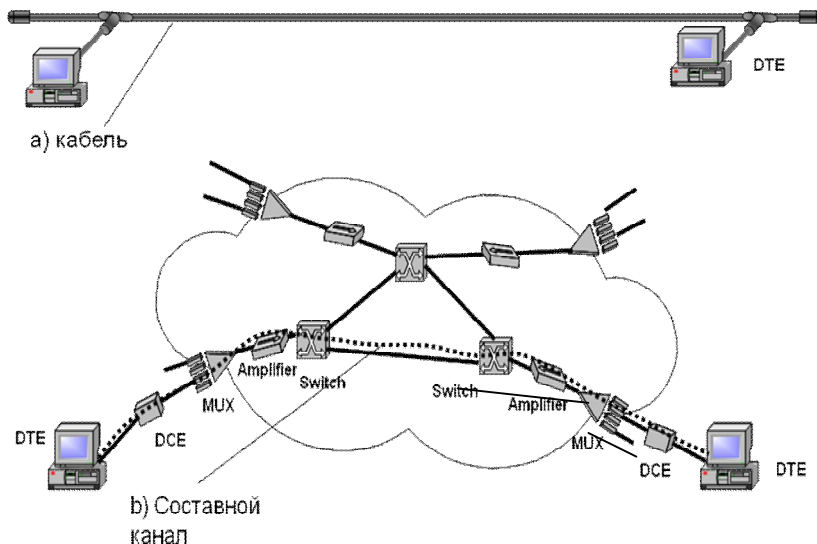


Рис. 2.1. Состав линии связи

В зависимости от способа представления информации электрическими сигналами различают аналоговые и цифровые каналы передачи данных. В аналоговых каналах для согласования параметров среды и сигналов применяют амплитудную, частотную, фазовую и квадротно-амплитудную

модуляции. В цифровых каналах для передачи данных используют самосинхронизирующиеся коды, а для передачи аналоговых сигналов – кодово-импульсную модуляцию.

2.2. Проблемы объединения нескольких компьютеров

2.2.1. Топология физических связей

При объединении в сеть более двух компьютеров, возникает проблема соединения их друг с другом, т. е. необходимость корректного выбора конфигурации связей, или топологии.

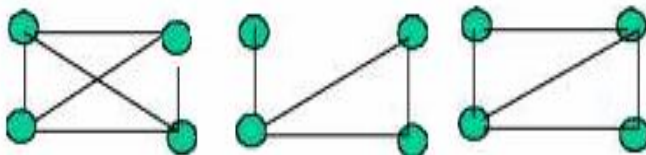
Под *топологией сети* понимается геометрическая фигура, вершинам которой, соответствуют конечные узлы сети (компьютеры) и коммуникационное оборудование (маршрутизаторы), а ребрам – физические или информационные связи между вершинами.

Виды коммуникационного оборудования: повторители, концентраторы, мосты, коммутаторы, маршрутизаторы, шлюзы.

Число возможных вариантов конфигураций резко возрастает при увеличении числа связываемых устройств. Например, три компьютера можно связать двумя способами (рис. 2.2, а), а для четырех можно применить уже шесть различных топологических конфигураций (если компьютеры неразличимы), (рис. 2.2, б).



а)



б)

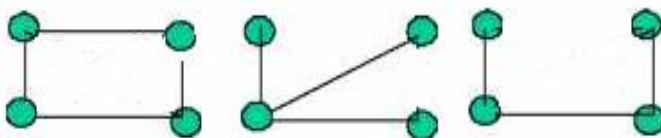


Рис. 2.2. Варианты связей компьютеров

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

Среди множества возможных конфигураций различают полносвязные и неполносвязные.

Полносвязные топологии в компьютерных сетях применяются редко. Чаще этот вид топологии используется в многомашинных комплексах или в глобальных сетях, объединяющих небольшое количество компьютеров.

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

Итак, выбор топологии сети чаще всего обусловлен:

- количеством компьютеров сети;
- расстоянием между компьютерами;
- необходимостью возможного расширения сети;
- задачами пользователей сети.

Самыми распространенными топологиями сети являются: точка-точка, звезда, общая шина, кольцо (рис. 2.3., 2.4).

Для каждого из типов топологий существуют свои особенности, достоинства и недостатки (табл. 2.1).

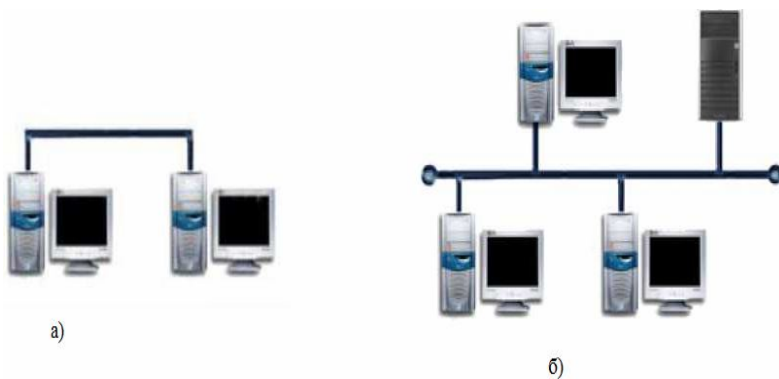


Рис. 2.3. Типовые топологии:
а) топология «Точка-точка»; б) топология «Общая шина»



Рис. 2.4. Типовые топологии:
а) топология «Кольцо»; б) топология «Звезда»

Таблица 2.1

Характеристики топологий

Характеристики	Топология
Шина	
Особенности	Характерен метод доступа с прослушиванием несущей частоты и обнаружением конфликта, который заключается в том, что узел сети прослушивает коммуникационный канал перед посылкой данных и посылает пакет только в том случае, если канал свободен. В случае если канал связи занят, попытка повторяется через определенный промежуток времени. При этом данные, переданные одним из узлов, посылаются во все остальные узлы, но распознаются и принимаются только тем, которому он был послан. В настоящее время практически не используется
Достоинства	Эффективное использование пропускной способности канала; невысокая стоимость проводки; простота разводки кабеля, реконфигурации и наращивания сети
Недостатки	Низкая надежность (незначительное повреждение кабеля выводит из строя всю сеть); невысокая производительность (в один момент времени только один компьютер может передавать данные); во время передачи по сети сигналы затухают, нужны повторители, усиливающие сигнал перед передачей в следующий сегмент
Кольцо	
Особенности	Данные передаются, как правило, в одном направлении. Данные, посланные от одного компьютера, совершив круговой оборот, возвращаются к источнику. Это бывает удобно при тестировании сети. Самый распространенный метод доступа - создание активного маркера, который движется по кольцу. Его дожидается компьютер, желающий передать данные по сети. С передаваемыми данными маркер продолжает круговое движение до адресата. Когда адресат принимает данные, маркер движется дальше до отправителя, который убеждается, что данные переданы, после чего маркер освобождается
Достоинства	Каждый из компьютеров усиливает передающийся сигнал; высокая устойчивость к перегрузкам; эффективная работа при передаче больших потоков

Продолжение табл. 2.1.

Недостатки	Не отличается большой надежностью, так как неисправность в одном из узлов разрывает всю сеть; время передачи велико, т. к. сигнал проходит последовательно через все компьютеры; подключение нового компьютера требует остановки всей сети; прокладка 2-го резервного кольца удорожает сеть
Звезда	
Особенности	Характерен метод доступа с передачей маркера. Он передается по кольцу в одном направлении. Каждый узел распознает передаваемый маркер. Если узел получает свободный маркер, то он может передать свои данные. Данные в маркере передаются всем компьютерам сети, но пакет распознает только тот компьютер, которому он был адресован, остальные его игнорируют. Имеется активный центр (это может быть и компьютер, и иное сетевое устройство – концентратор (hub, хаб), который управляет информационным обменом). В современных сетях имеет наибольшее распространение
Достоинства	Большая надежность, так как вывести из строя всю сеть может только неисправность концентратора; отказ периферийного компьютера или обрыв связи с ним не влияет на работу сети; хаб выполняет функции повторителя, восстанавливая сигнал; лёгкость при обслуживании; в сети концентратор может блокировать информацию, запрещенную администратором к передаче
Недостатки	Сложность наращивания сети, так как она может быть ограничена количеством портов концентратора; выход из строя центрального компьютера приводит к отказу сети

2.2.2. Адресация узлов сети

Второй проблемой, которую нужно учитывать при объединении трех и более компьютеров, является проблема их адресации, точнее адресации их сетевых интерфейсов. Иногда вместо точного выражения *«адрес сетевого интерфейса»* используют упрощенное – *«адрес узла сети»*. Один компьютер может иметь несколько сетевых интерфейсов. Например, для создания полносвязной структуры из N компьютеров необходимо,

чтобы у каждого из них имелся N-1 интерфейс. По количеству адресуемых интерфейсов, адреса можно классифицировать:

- уникальный адрес (*unicast*), используется для идентификации отдельных интерфейсов;
- групповой адрес (*multicast*), идентифицирует сразу несколько интерфейсов, поэтому данные, помеченные групповым адресом, доставляются каждому из узлов, входящих в группу;
- данные, направленные по широковещательному адресу (*broadcast*), должны быть доставлены всем узлам сети;
- в новой версии протокола IPv6 определен адрес произвольной рассылки (*anycast*), который, так же как и групповой адрес, задает группу адресов, однако данные, посланные по этому адресу, должны быть доставлены не всем адресам данной группы, а любому из них.

Адреса могут быть числовыми (*129.26.255.255* или *81.la.ff.ff*) и символьными (*site.domen.ru*, *willi-winki*).

Символьные адреса (имена) предназначены для запоминания людьми и несут смысловую нагрузку. Для работы в больших сетях символьное имя может иметь иерархическую структуру, например, *ftp-arch1.ucl.ac.uk*. Хотя символьные имена удобны для людей, из-за переменного формата и потенциально большой длины их передача по сети не очень экономична.

Множество всех адресов, которые являются допустимыми в рамках некоторой схемы адресации, называется адресным пространством.

Адресное пространство может иметь плоскую (*линейную*) организацию или *иерархическую* организацию.

При плоской организации множество адресов никак не структурировано. Примером плоского числового адреса является MAC-адрес, предназначенный для однозначной идентификации сетевых интерфейсов в локальных сетях. Такой адрес используется только аппаратурой, поэтому его стараются сделать по возможности компактным и записывают в виде двоичного или шестнадцатеричного числа, например *0081005e24a8*. При задании MAC-адресов не требуется выполнение ручной работы, они встраиваются в аппаратуру компаний – изготовителем, поэтому их называют также

аппаратными адресами (hardware addresses). Использование плоских адресов является жестким решением – при замене аппаратуры, например, сетевого адаптера, изменяется и адрес сетевого интерфейса компьютера.

В случае иерархической организации адресное пространство организовано в виде вложенных друг в друга подгрупп, которые, последовательно сужают адресуемую область и определяют отдельный сетевой интерфейс.

В трехуровневой структуре адресного пространства адрес конечного узла задается тремя составляющими: *идентификатором группы (K)*, в которую входит данный узел, *идентификатором подгруппы (L)* и, *идентификатором узла (n)*, однозначно определяющим его в подгруппе. Иерархическая адресация во многих случаях оказывается более рациональной, чем плоская. Иерархическая система адресации позволяет при перемещении данных до определенного момента пользоваться только старшей составляющей адреса (например, идентификатором группы K), затем для дальнейшей локализации адресата задействовать следующую по старшинству часть (L) и, в конечном счете, младшую часть (n). Типичными представителями иерархических числовых адресов являются сетевые 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 после доставки сообщения в нужную сеть; точно так же, как название улицы используется почтальоном, только после того, как письмо доставлено в нужный город.

На практике применяют сразу несколько схем адресации, так что сетевой интерфейс компьютера может одновременно иметь несколько адресов-имен. Каждый адрес применяется в случае, когда наиболее удобен соответствующий вид адресации. Для преобразования адресов из одного вида в другой используются специальные вспомогательные протоколы, называемые протоколами разрешения адресов

Конечной целью данных, пересылаемых по сети, являются не сетевые интерфейсы или компьютеры, а выполняемые на этих устройствах программы – процессы. Поэтому в адресе

назначения наряду с информацией, идентифицирующей интерфейс устройства, должен указываться адрес процесса, которому предназначены посылаемые по сети данные. Поэтому достаточно обеспечить уникальность адреса процесса в пределах компьютера. Примером адресов процессов являются номера портов протоколов *TCP* и *UDP*, используемые в стеке *TCP/IP*.

2.2.3. Коммутация, маршрутизация

После решения вопроса с топологией и адресацией, возникает самая важная проблема: выбор способа передачи данных между конечными узлами. Если топология сети неполносвязная, то обмен данными между произвольной парой конечных узлов (абонентов) должен идти в общем случае через транзитные узлы. Например, в сети на рис. 2.5, узлы 2 и 4, не связанные непосредственно вынуждены передавать данные через транзитные узлы, в качестве которых могут выступать, например, узлы 1 и 5. Узел 1 должен выполнить передачу данных с интерфейса «А» на интерфейс «В», а узел 5 – с интерфейса «F» на «B». Последовательность транзитных узлов (сетевых интерфейсов) на пути от отправителя к получателю называется *маршрутом*.

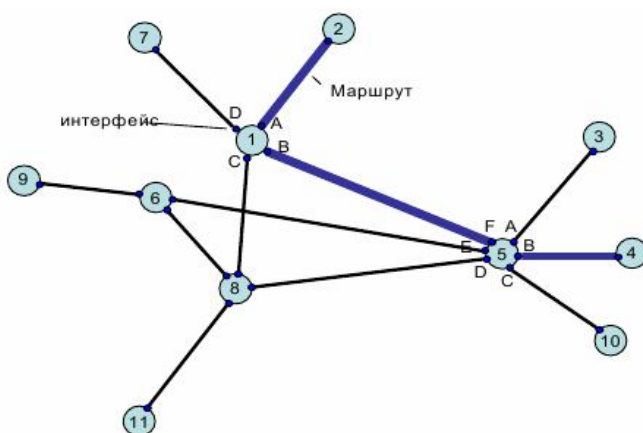


Рис. 2.5. Коммутация абонентов через сеть транзитных узлов

В самом общем виде задача соединения конечных узлов через сеть транзитных узлов называется *задачей коммутации*. Она может быть представлена в виде нескольких взаимосвязанных частных задач:

1. Определение информационных потоков, для которых требуется прокладывать пути;
2. Определение маршрутов для потоков;
3. Сообщение о найденных маршрутах узлам сети;
4. Продвижение потоков, т. е. распознавание потоков и их локальная коммутация на каждом транзитном узле;
5. Мультиплексирование и демultipлексирование потоков.

Маршрутизация и типы коммутации. Задача маршрутизации включает в себя две подзадачи:

- определение маршрута;
- оповещение сети о выбранном маршруте.

Определить маршрут – это значит выбрать последовательность транзитных узлов и их интерфейсов, через которые надо передавать данные, чтобы доставить их адресату.

Маршрут может определяться *эмпирически* («вручную») администратором сети на основании различных, часто не формализуемых соображений. Среди побудительных мотивов выбора пути могут быть: особые требования к сети со стороны различных типов приложений, решение передавать трафик через сеть определенного поставщика услуг, предположения о пиковых нагрузках на некоторые каналы сети, соображения безопасности.

Но эмпирический подход к определению маршрутов для большой сети, со сложной топологией, не пригоден. Для этого случая используются автоматические методы определения маршрутов. При этом конечные узлы и другие устройства сети оснащаются специальными программными средствами, которые организуют взаимный обмен служебными сообщениями, позволяющий каждому узлу составить свое «представление» о сети. Затем на основе собранных данных программными методами определяются рациональные маршруты.

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. В качестве *критериев выбора* могут выступать, например:

- номинальная пропускная способность;
- загруженность каналов связи;
- задержки, вносимые каналами;
- количество промежуточных транзитных узлов;
- надежность каналов и транзитных узлов.

Маршрутизация – процесс определения в сети пути, по которому блок данных может достигнуть адресата. Основная задача маршрутизации – переключение трафика. Переключение – это процесс приема сообщения, выбора подходящего маршрута дальнейшего следования и отправка его по этому маршруту. Данная операция обслуживается четырьмя различными процессами: входным драйвером, процессом выбора маршрута, очередью и выходным драйвером. Маршрутизация выполняется на основе данных, содержащихся в таблице маршрутов. Строка в таблице маршрутов состоит из полей: адрес сети назначения; адрес следующего маршрутизатора (т. е. узла, который знает, куда дальше отправить дейтаграмму, адресованную в сеть назначения); вспомогательные поля.

Маршрутизация решает две задачи:

- выбор оптимального, по некоторому критерию, пути продвижения информации от источника к пункту назначения через объединенную сеть;
- транспортировка информационных блоков (пакетов) по выбранному маршруту, или коммутация.

Чтобы по адресу сети назначения можно было бы выбрать рациональный маршрут дальнейшего следования пакета, каждый конечный узел и маршрутизатор анализируют специальную информационную структуру, которая называется *таблицей маршрутизации*.

Основу любой сетевой технологии в своей совокупности составляет комплекс технических решений обобщенной задачи

коммутации. К частным задачам относятся: определение потоков и соответствующих маршрутов; фиксация маршрутов в конфигурационных параметрах и таблицах сетевых устройств; распознавание потоков и передача данных между интерфейсами одного устройства; мультиплексирование/демультиплексирование потоков; разделение среды передачи.

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

При решении проблемы коммутации абонентов в сетях часто выделяют два основополагающих метода:

- коммутация каналов (*circuit switching*);
- коммутация пакетов (*packet switching*).

Внешне обе эти схемы соответствуют структуре сети (рис. 2.6), однако возможности и свойства их различны.

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

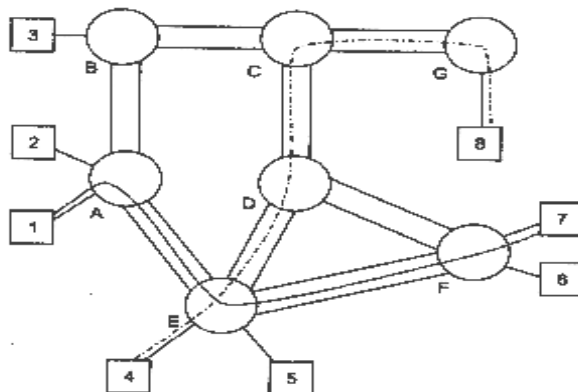


Рис. 2.6. Общая структура сети с коммутацией абонентов

Выводы:

Среды передачи данных могут быть общего пользования или выделенными для конкретного пользователя. В качестве линий передачи данных выступают: коаксиальный кабель, витая пара проводов, световод и т. д.

Полоса пропускания связана со скоростью передач информации. Различают бодовую (модуляционную) и информационную скорости. Два метода разделения линии передачи данных: временное мультиплексирование (Time Division Multiplexing, TDM), при котором каждому каналу выделяется некоторый квант времени, и частотное разделение (FDM, Frequency Division Method), при котором каналу выделяется некоторая полоса частот.

Для объединения в сеть более двух компьютеров возникают проблемы: выбора топологии; способа адресации; способа разделения линий связи; механизма коммутации.

В неполносвязных сетях соединение пользователей осуществляется путем коммутации через сеть транзитных узлов. При этом должны быть решены следующие задачи: определение потоков данных и маршрутов для них, продвижение данных в каждом транзитном узле, мультиплексирование и демультимплексирование потоков.

По количеству адресуемых интерфейсов, адреса

подразделяются на: уникальный адрес (*unicast*), используемый для идентификации отдельных интерфейсов; групповой адрес (*multicast*), идентифицирующий сразу несколько интерфейсов; широковещательный адрес (*broadcast*); адрес произвольной рассылки (*anycast*).

Маршрутизация решает задачи: выбор оптимального пути продвижения информации от источника к пункту назначения через объединенную сеть; транспортировку пакетов по выбранному маршруту или коммутацию. Коммутация абонентов в сетях выполняется с помощью методов: коммутация каналов (*circuit switching*); коммутация пакетов (*packet switching*).

Контрольные вопросы:

1. Что представляет собой среда передачи?
2. Что представляют собой линии передачи данных?
3. Дайте определение понятию «топология».
4. К какому типу топологии можно отнести структуру, образованную тремя связанными друг с другом узлами (в виде треугольника)?
5. К какому типу топологии можно отнести структуру, образованную четырьмя связанными друг с другом узлами (в виде квадрата)?
6. К какому типу топологии можно отнести структуру, образованную тремя последовательно соединенными друг с другом узлами (последний не связан с первым)?
7. Какая скорость связана с полосой пропускания?
8. Объясните понятие «бодовая скорость».
9. Дайте определение понятию «канал связи».
10. В чем заключается отличие между методами Time Division Multiplexing и Frequency Division Method.
11. Какими особенностями отличаются топологии: «звезда», «общая шина», «кольцо»?
12. Какие виды адресов используются при объединении компьютеров в сеть?
13. В чем заключается линейная организация адресного пространства?

14. В чем заключается иерархическая организация адресного пространства?
15. Какими составляющими задается адрес конечного узла в трехуровневой структуре адресного пространства?
16. Какие задачи решаются при соединении конечных узлов?
17. Какие критерии выбора применяются при выборе наилучшего маршрута?
18. Какие задачи решает маршрутизация?
19. В чем отличие методов коммутации каналов и коммутации пакетов?

3. КОММУТАЦИЯ КАНАЛОВ И ПАКЕТОВ

3.1. Коммутация каналов

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

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

Например, если сеть (рис. 2.6), работает по технологии коммутации каналов, то узел 1, для передачи данных узлу 7, прежде всего, должен передать специальный запрос на установление соединения коммутатору «А», указав адрес назначения 7. Коммутатор «А» должен выбрать маршрут образования составного канала, а затем передать запрос следующему коммутатору, в данном случае «Е». Далее коммутатор «Е» передаст запрос коммутатору «F», а тот, в свою очередь, передаст запрос узлу 7. Если узел 7 принимает запрос на установление соединения, он направляет по уже установленному каналу ответ исходному, после чего составной канал считается скоммутированным, и узлы 1 и 7 могут обмениваться по нему данными, например, вести телефонный разговор.

Техника коммутации каналов имеет свои достоинства и недостатки.

Основные достоинства:

1. Постоянная и известная скорость передачи данных по установленному между конечными узлами каналу позволяет пользователю сети на основе заранее произведенной оценки требуемой для качественной передачи данных пропускной способности установить в сети канал нужной скорости;

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

К недостаткам относятся следующее:

1. Отказ сети в обслуживании запроса на установление соединения. Такая ситуация может сложиться из-за того, что на некотором участке сети соединение нужно установить вдоль некоторого физического канала, через который уже проходит максимальное количество информационных потоков, для данной техники мультиплексирования данного канала;

2. Нерациональное использование пропускной способности физических каналов. После установления соединения часть пропускной способности отводится составному каналу на все время соединения, т. е. до тех пор, когда соединение не будет разорвано по инициативе абонентов или самой сети. В то же время во многих случаях абонентам не нужна пропускная способность канала на все время соединения, например, в телефонном разговоре встречаются паузы, еще более неравномерным во времени является взаимодействие компьютеров. Невозможность динамического перераспределения пропускной способности физического канала является принципиальным ограничением сети с коммутацией пакетов, т. к. единицей коммутации здесь является информационный поток в целом. Модификация методов коммутации для снятия этого ограничения приводит к технике коммутации пакетов;

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

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

3.2. Коммутация пакетов

3.2.1. Принципы коммутации пакетов

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

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

При коммутации пакетов все передаваемые пользователем сети сообщения разбиваются в исходном узле на сравнительно небольшие части, называемые пакетами. *Сообщением* называется логически завершенная порция данных – запрос на передачу файла, ответ на этот запрос, содержащий весь и т. п. Сообщения могут иметь произвольную длину, от нескольких

байт до многих мегабайт. Пакеты обычно тоже могут иметь переменную длину, но в пределах, например от 46 до 1500 байт. Каждый пакет снабжается заголовком, в котором указывается адресная информация, необходимая для доставки пакета узлу назначения, а также номер пакета, который будет использоваться узлом назначения для сборки сообщения (рис. 3.1). Пакеты транспортируются в независимые информационные блоки.

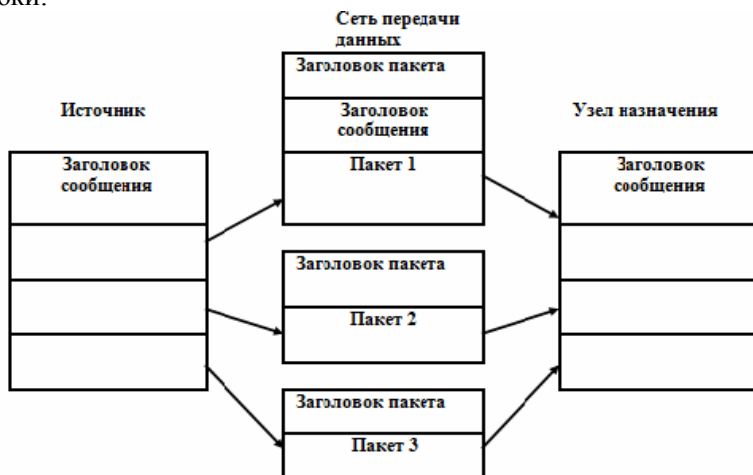


Рис. 3.1. Разбиение сообщений на пакеты

Коммутаторы сети принимают пакеты от конечных узлов и на основании адресной информации передают их другу, и в конечном итоге – узлу назначения. Коммутаторы пакетной сети отличаются от коммутаторов каналов тем, что используют внутреннюю буферную память для временного хранения пакетов, когда входной порт коммутатора в момент принятия пакета занят передачей другого пакета (рис. 3.2). В этом случае пакет находится некоторое время в очереди пакетов в буферной памяти выходного порта, а когда до него дойдет очередь, то он передается следующему коммутатору. Такая схема передачи данных позволяет сглаживать пульсации трафика на магистральных связях между коммутаторами и тем самым использовать их наиболее эффективным образом для

повышения пропускной способности сети в целом.

Конечно, для пары абонентов наиболее эффективным было бы предоставление им в единоличное пользование коммутированного канала связи, как это делается в сетях с коммутацией каналов. При этом способе время взаимодействия пары абонентов было бы минимальным, так как данные без задержек передавались бы от одного абонента к другому. Простой канала во время пауз передачи абонентов не интересуют, для них важно быстрее решить свою собственную задачу. Сеть с коммутацией пакетов замедляет процесс взаимодействия этой пары абонентов, т. к. их пакеты могут находиться в режиме ожидания, потому как в коммутаторах, по магистральным связям могут передаваться другие пакеты, пришедшие в коммутатор ранее. Тем не менее, общий объем передаваемых сетью компьютерных данных в единицу времени при коммутации пакетов выше, чем при коммутации каналов.

Это происходит потому, что пульсации отдельных абонентов в соответствии с законом больших чисел распределяются во времени так, что их пики не совпадают. Поэтому коммутаторы постоянно и достаточно равномерно загружены работой, если число обслуживаемых ими абонентов действительно велико. На рис. 3.2 показано, что трафик, поступающий от конечных узлов на коммутаторы, очень неравномерно распределен во времени. Однако коммутаторы более высокого уровня иерархии, которые обслуживают соединения между коммутаторами нижнего уровня, загружены более равномерно, и поток пакетов в магистральных каналах, соединяющих коммутаторы верхнего уровня, имеет почти максимальный коэффициент использования. Буферизация сглаживает пульсации, поэтому коэффициент пульсации на магистральных каналах гораздо ниже, чем на каналах абонентского доступа – он может быть равным 1:10 или даже 1:2.

Более высокая эффективность сетей с коммутацией пакетов по сравнению с сетями с коммутацией каналов (при равной пропускной способности каналов связи) была доказана в 60-е годы как экспериментально, так и путем имитационного моделирования. Здесь уместна аналогия с мультипрограммными операционными системами. Каждая отдельная программа в

такой системе выполняется дольше, чем в однопрограммной системе, когда программе выделяется все процессорное время, пока она не завершит свое выполнение. Однако общее число программ, выполняемых за единицу времени в мультипрограммной системе больше, чем в однопрограммной.

Достоинства сетей с коммутацией пакетов:

1. Высокая общая пропускная способность сети при передаче пульсирующего трафика;
2. Возможность динамического перераспределения пропускной способности физических каналов связи между абонентами в соответствии с реальными потребностями их трафика.

Недостатками сетей с коммутацией пакетов являются:

1. Неопределенность скорости передачи данных между абонентами сети, обусловленная зависимостью задержек в очередях буферов коммутаторов сети от общей загрузки сети;
2. Переменная величина задержки пакетов данных, которые могут достигать значительных величин в моменты мгновенных перегрузок сети;
3. Возможные потери данных из-за переполнения буферов.

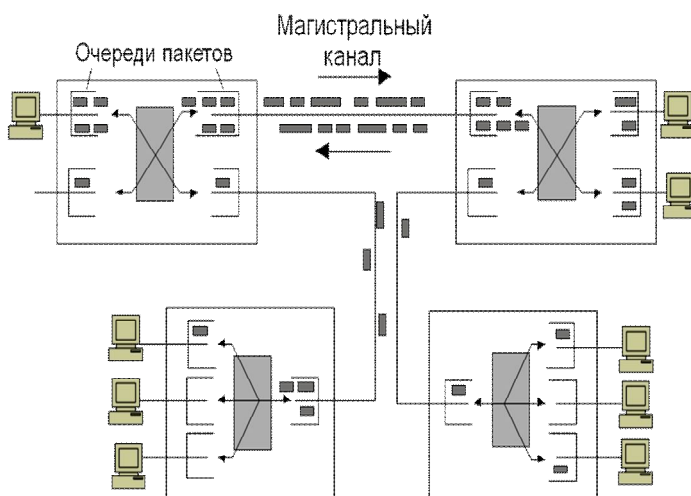


Рис. 3.2. Сглаживание пульсаций графика в сети с коммутацией пакетов

Существуют два класса сетей с коммутацией пакетов: с виртуальным каналом; дейтаграммные (Internet на базе IP, X.25, Frame Relay, ATM).

3.2.2. Виртуальный канал

Виртуальный канал устанавливается перед началом соединения и является единственным маршрутом для данного соединения. Динамический виртуальный канал устанавливается путем отправки запроса на соединение, которое проходит через коммутаторы и прокладывает виртуальный канал. При этом коммутаторы запоминают маршрут и именно по нему пересылают все последующие пакеты. Статический виртуальный канал создается администраторами, путем ручной настройки.

При отказе одного из элементов маршрута, соединение разрывается, и виртуальный канал нужно прокладывать заново (основное отличие от дейтаграммных сетей).

Дейтаграммные сети аналогичны обычной почте. Пакет помещается в своеобразный конверт (заголовок пакета содержит адрес назначения), который имеет иерархическую структуру. По прибытии пакета на маршрутизатор, он сравнивает соответствующую часть адреса назначения и отправляет пакет к следующему маршрутизатору. Отметим: коммутация с виртуальным каналом поддерживает, а дейтаграммные сети – не поддерживают информацию о маршруте любого соединения. В дейтаграммных сетях коммутатор принимает решение о маршруте каждого пакета одновременно.

Сети с коммутацией пакетов, в которых реализованы методы обеспечения качества обслуживания, позволяют одновременно передавать различные виды трафика, в т. ч. телефонный и компьютерный. Поэтому методы коммутации пакетов сегодня считаются наиболее перспективными для построения конвергентной сети, обеспечивающей комплексные качества услуг для абонентов любого типа. Возможно появление новых технологий коммутации, в том или ином виде комбинирующих принципы коммутации пакетов и каналов.

3.3. Коммутация сообщений

Коммутация сообщений по своим принципам близка к коммутации пакетов. Метод коммутации сообщений не предполагает предварительного установления выделенного пути между станциями. Под *коммутацией сообщений* понимается передача единого блока данных между транзитными компьютерами сети с временной буферизацией этого блока на диске каждого компьютера (рис. 3.3). При передаче сообщения станция добавляет к нему адрес назначения. Сообщение передается по сети от узла к узлу. Сообщение в отличие от пакета имеет произвольную длину, определяемая не технологическими соображениями, а содержанием информации, составляющей сообщение. Например, сообщением может быть текстовый документ, файл с кодом программы, электронное письмо и т. д.



Рис. 3.3. Коммутация сообщений

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

По такой схеме обычно передаются сообщения, не требующие немедленного ответа, чаще всего сообщения электронной почты. Режим передачи с промежуточным хранением на диске называется режимом *хранения и передачи*. Такой тип сети называют «*сеть с промежуточным хранением*»

(*store-and-forward network*).

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

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

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

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

3.4. Постоянная и динамическая коммутация

Как сети с коммутацией пакетов, так и сети с коммутацией каналов можно разделить на два класса: на сети с *динамической коммутацией* и сети с *постоянной коммутацией*.

В первом случае сеть разрешает устанавливать соединение по инициативе пользователя сети. Коммутация выполняется на время сеанса связи, а затем (по инициативе одного из взаимодействующих пользователей) связь разрывается. В общем случае любой пользователь сети может соединиться с любым другим пользователем сети. Обычно период соединения между парой пользователей при динамической коммутации составляет

от нескольких секунд до нескольких часов и завершается при выполнении определенной работы – передачи файла, просмотра страницы текста или изображения и т. п.

Во втором случае сеть разрешает паре пользователей заказать соединение на длительный период времени. Соединение устанавливается не пользователями, а персоналом, обслуживающим сеть. Время, на которое устанавливается постоянная коммутация, измеряется обычно несколькими месяцами. Режим *постоянной* коммутации в сетях с коммутацией каналов часто называется сервисом *выделенных (dedicated)*, или *арендуемых (leased)*, каналов. В том случае, когда постоянное соединение через сеть коммутаторов устанавливается с помощью автоматических процедур, инициированных обслуживающим персоналом, его часто называют *полупостоянным (semi-permanent)* соединением, в отличие от режима ручного конфигурирования каждого коммутатора.

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

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

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

Выводы:

В сетях с коммутацией каналов создается непрерывный информационный канал по запросу пользователя, образуемый путем резервирования «цепочки» линий связи, соединяющих абонентов на время передачи данных. На всем своем протяжении канал передает данные с одной и той же скоростью. Поэтому через сеть с коммутацией каналов можно качественно передавать данные, чувствительные к задержкам (голос, видео). Но невозможность динамического перераспределения пропускной способности физического канала является принципиальным

недостатком этой сети, поэтому она неэффективна для передачи пульсирующего компьютерного трафика.

При коммутации пакетов передаваемые данные разбивают в исходном узле на небольшие части – пакеты. Пакет снабжается заголовком, с указанием адреса назначения, поэтому он может обрабатываться коммутатором независимо от остальных данных. Коммутация пакетов повышает производительность сети при передаче пульсирующего трафика, т. к. при обслуживании большого числа независимых потоков периоды их активности не всегда совпадают во времени. Пакеты поступают в сеть без предварительного резервирования ресурсов в том темпе, в котором их генерирует источник. Но этот способ также имеет отрицательные стороны: задержки передачи носят случайный характер, поэтому возникают проблемы при передаче трафика реального времени.

В сетях с коммутацией пакетов может использоваться один из трех алгоритмов продвижения пакетов: дейтаграммная, с установлением логического соединения и с установлением виртуального канала.

Контрольные вопросы:

1. Какие типы мультиплексирования и коммутации используются в телефонных сетях?
2. Какие свойства сетей с коммутацией каналов свидетельствуют об их недостатках?
3. Используется ли буферизация в сетях с коммутацией каналов?
4. Учитывается ли в дейтаграммных сетях существование потоков данных?
5. Дайте определение логического соединения.
6. Какое логическое соединение может быть названо виртуальным каналом?
7. Какие данные относятся к трафику реального времени?
8. Какая схема передачи данных позволяет сглаживать пульсации трафика?
9. Какой тип сети называют «сеть с промежуточным хранением»?
10. В чем заключается отличие между постоянной и динамической коммутациями?

4. АРХИТЕКТУРА И СТАНДАРТИЗАЦИЯ СЕТЕЙ

Архитектура предполагает описание сети в виде системы элементов, выполняющих определенную частную функцию, при этом все элементы в совокупности согласованно решают общую задачу взаимодействия компьютеров. Иначе архитектура сети демонстрирует декомпозицию общей задачи взаимодействия компьютеров на отдельные подзадачи, решаемые отдельными элементами сети. Одним из важных элементов архитектуры сети является коммуникационный *протокол* – формализованный набор правил взаимодействия узлов сети.

Модель OSI является международным стандартом, определяющим способ декомпозиции задачи взаимодействия «по вертикали», поручая эту задачу коммуникационным протоколам семи уровней. Уровни образуют иерархию, известную как стек протоколов, где каждый вышестоящий уровень использует нижестоящий в качестве удобного инструмента для решения своих задач.

Стандартная архитектура компьютерной сети определяет также распределение протоколов между элементами сети – конечными узлами (компьютерами) и промежуточными узлами (повторителями, концентраторами, мостами, коммутаторами, маршрутизаторами, шлюзами).

Промежуточные узлы выполняют только транспортные функции стека протоколов, передавая трафик между конечными узлами.

Конечные узлы поддерживают весь стек протоколов, предоставляя информационные услуги, например Web-сервис. Такое распределение функций означает смещение «интеллекта» сети на ее периферию.

4.1. Декомпозиция задачи сетевого взаимодействия

Организация взаимодействия между устройствами сети – это сложная задача, для решения которых используется известный универсальный прием – *декомпозиция*, разбиение одной сложной задачи на несколько более простых задач-модулей. Декомпозиция состоит в четком определении:

- функций каждого модуля;
- порядка их взаимодействия (межмодульных интерфейсов).

Итогом такого логического упрощения задачи является появление возможностей: независимого тестирования; разработки и модификации модулей.

Главная цель, преследуемая при объединении компьютеров в информационную систему, – это возможность использования ресурсов каждого компьютера всеми пользователями системы. Для реализации такой возможности, необходимо обеспечить компьютеры средствами взаимодействия с другими компьютерами системы. Задача разделения системных ресурсов включает в себя решение множества проблем: выбор способа адресации компьютеров и согласование электрических сигналов при установлении электрической связи; обеспечение надежной передачи данных и обработка сообщений об ошибках; формирование отправляемых и интерпретация полученных сообщений, и много других задач.

Для решения применяется декомпозиция на несколько частных проблем – подзадач. Каждой подзадаче назначается некоторый модуль. При этом четко определяются функции каждого модуля и правила их взаимодействия. Частным случаем декомпозиции задачи является многоуровневое представление, при котором все множество модулей, решающих подзадачи, разбивается на иерархически упорядоченные группы – уровни. Для каждого уровня определяется набор функций-запросов, с которыми к модулям данного уровня могут обращаться модули вышележащего уровня для решения своих задач. Такой формально определенный набор функций, выполняемых данным уровнем для вышележащего уровня, а также форматы сообщений, которыми обмениваются два соседних уровня в ходе своего взаимодействия, называется *интерфейсом*.

Межуровневый интерфейс или **интерфейс услуг** определяет набор функций, предоставляемых нижележащим уровнем вышележащему уровню.

Многоуровневое представление средств сетевого взаимодействия имеет свою специфику, связанную с тем, что в процессе обмена сообщениями участвуют, по меньшей мере, *две стороны*, т. е. в данном случае необходимо организовать

согласованную работу двух иерархий аппаратных и программных средств, работающих на разных компьютерах. При передаче сообщений оба участника сетевого обмена должны принять множество соглашений. Например, они должны согласовать уровни и форму электрических сигналов, способ определения длины сообщений, договориться о методах контроля достоверности и т. п. Другими словами, соглашения должны быть приняты для всех уровней, начиная от самого низкого уровня передачи битов, до самого высокого уровня, реализующего обслуживание пользователей сети.

С каждой стороны средства взаимодействия представлены четырьмя уровнями, каждый из которых поддерживает интерфейсы двух типов:

- интерфейсы услуг с выше- и нижележащим уровнями «своей» иерархии средств;

- интерфейс со средствами взаимодействия другой стороны, расположенными на том же уровне иерархии. Этот тип интерфейса называют *протоколом*.

Другими словами, **протокол** – это совокупность соглашений относительно способа представления данных, обеспечивающего их передачу в нужных направлениях и правильную интерпретацию данных всеми участниками процесса информационного обмена.

Итак, понятия «интерфейс» и «протокол», в сущности, обозначают одно и то же, а именно – формализованное описание процедуры взаимодействия двух объектов, но в сетях за ними закрепили разные области действия: *протоколы* определяют правила взаимодействия модулей одного уровня в разных узлах, а *интерфейсы* – правила взаимодействия модулей соседних уровней в одном узле.

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

Протоколы нижних уровней часто реализуются комбинацией программных и аппаратных средств, а протоколы верхних уровней, как правило, программными средствами.

Программный модуль, реализующий некоторый протокол, называют протокольной сущностью, или, для краткости, протоколом.

Протокольные сущности одного уровня двух взаимодействующих сторон обмениваются сообщениями в соответствии с определенным для них протоколом. Сообщения состоят из заголовка и поля данных (иногда оно может отсутствовать). Обмен сообщениями является своеобразным языком общения, с помощью которого каждая из сторон «объясняет» другой стороне, что необходимо сделать на каждом этапе взаимодействия. Работа каждого протокольного модуля состоит в чтении, анализе заголовков поступающих к нему сообщений и выполнении связанных с этим действий. Заголовки сообщений разных протоколов имеют разную структуру, что соответствует различиям в их функциональности. Чем сложнее структура заголовка сообщения, тем более сложные функции возложены на соответствующий протокол.

4.2. Модель OSI

Поскольку информационный обмен – процесс многофункциональный, то протоколы делятся на уровни. К каждому уровню относится группа родственных функций. Для правильного взаимодействия узлов различных локальных сетей их архитектура должна быть открытой. Этим целям служат унификация и стандартизация в области телекоммуникаций и локальных сетей. Унификация и стандартизация протоколов выполняется рядом международных организаций, что наряду с разнообразием типов сетей породило большое число различных протоколов. Широко распространенными являются протоколы, разработанные для сети ARPANET и применяемые в глобальной сети Internet, протоколы открытых систем Международной организации по стандартизации (*ISO – International Standard Organization*), протоколы Международного телекоммуникационного союза (*International Telecommunication Union – ITU*) и протоколы Института инженеров по электротехнике и электронике (*IEEE – Institute of Electrical and Electronics Engineers*).

В начале 80-х годов ряд международных организаций по стандартизации, в частности *ISO*, а также *ITU* – разработали стандартную модель взаимодействия открытых систем (*Open System Interconnection, OSI*).

Модель *OSI* определяет:

- уровни взаимодействия систем в сетях с коммутацией пакетов;
- стандартные названия уровней;
- функции, которые должен выполнять каждый уровень.

Модель *OSI* не содержит описаний реализаций конкретного набора протоколов. Каждый уровень имеет дело с совершенно определенным аспектом взаимодействия сетевых устройств.

Модель *OSI* описывает только системные средства взаимодействия, реализуемые операционной системой, системными программными и аппаратными средствами. Модель не включает средства взаимодействия приложений конечных пользователей.

Модель признана всеми международными организациями как основа для стандартизации протоколов информационных сетей. Информационная сеть рассматривается как совокупность функций, подразделяемая на группы, называемые *уровнями*. Разделение на уровни позволяет вносить изменения в средства реализации одного уровня без перестройки средств других уровней, что значительно упрощает и удешевляет модернизацию средств по мере развития техники. Модель содержит семь уровней: *прикладной (Application)*, *представительный (Presentation)*, *сеансовый (Session)*, *транспортный (Transport)*, *сетевой (Network)*, *канальный (Link, уровень звена данных)*, *физический (Physical)*. Каждый уровень поддерживает интерфейсы с выше- и нижележащими уровнями (рис. 4.1).

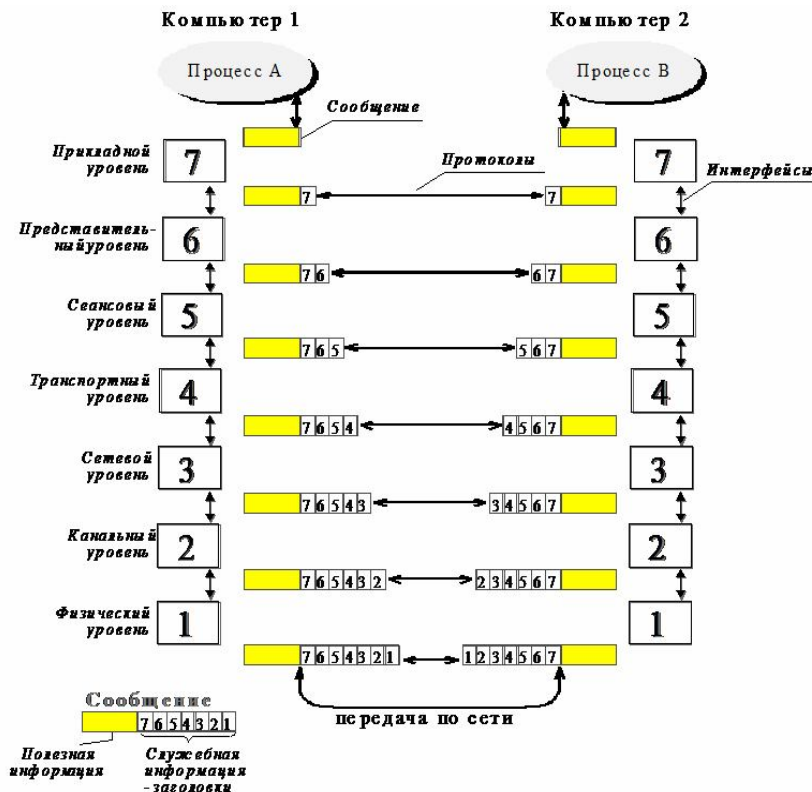


Рис. 4.1. Модель взаимодействия открытых систем ISO/OSI

На физическом уровне осуществляется передача битов по физическим каналам, таким, например, как витая пара, коаксиальный или оптоволоконный кабель. К этому уровню имеют отношение характеристики физических сред передачи данных (полоса пропускания, помехозащищенность, волновое сопротивление и другие) и электрических сигналов (требования к фронтам импульсов, уровням напряжения или тока передаваемого сигнала, тип кодирования, скорость передачи сигналов). Кроме этого, здесь стандартизируются 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.

Одной из задач **канального уровня** является проверка доступности среды передачи. Другой задачей является реализация механизмов обнаружения и коррекции ошибок. Для этого на канальном уровне биты группируются в наборы, называемые кадрами (*frames*). Канальный уровень обеспечивает корректность передачи каждого кадра, помещая специальную последовательность бит в начало и конец каждого кадра, чтобы отметить его, а также вычисляет контрольную сумму, суммируя все байты кадра определенным способом и добавляя контрольную сумму к кадру. Когда кадр приходит, получатель снова вычисляет контрольную сумму полученных данных и сравнивает результат с контрольной суммой из кадра. Если они совпадают, кадр считается правильным и принимается. Если же контрольные суммы не совпадают, то фиксируется ошибка.

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 для топологий: общая шина, кольцо и звезда. Примерами протоколов канального уровня являются протоколы Ethernet, Token Ring и др. 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.

Сетевой уровень служит для образования единой транспортной системы, объединяющей несколько подсетей с различными принципами передачи информации между конечными узлами. Внутри подсети доставка данных регулируется канальным уровнем, а вот доставкой данных между подсетями занимается сетевой уровень. Сообщения сетевого уровня принято называть *пакетами* (*packets*). При организации доставки пакетов на сетевом уровне используется понятие «номер сети». В этом случае адрес получателя состоит из номера сети и номера компьютера в этой сети.

Работа **транспортного уровня** заключается в обеспечении 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. Как правило, все протоколы, начиная с транспортного уровня и выше, реализуются программными средствами конечных узлов сети – компонентами их сетевых операционных систем.

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

Уровень представления гарантирует то, что информация, передаваемая прикладным уровнем, будет понятна прикладному уровню в другой системе. При необходимости выполняется преобразование форматов данных в некоторый общий формат представления, а на приеме, соответственно, выполняется обратное преобразование. На этом уровне может выполняться шифрование и дешифрование данных, благодаря которому секретность обмена данными обеспечивается сразу для всех прикладных сервисов.

Прикладной уровень – это в действительности просто набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые Web-страницы, а также организуют свою совместную работу, например, с помощью протокола электронной почты. Единица данных, которой оперирует прикладной уровень, обычно называется *сообщением (message)*.

В конкретных случаях может возникать потребность в реализации лишь части названных функций, тогда соответственно в сети имеется лишь часть уровней. Модель *OSI* описывает только системные средства взаимодействия, не касаясь приложений конечных пользователей. Приложения реализуют свои собственные протоколы взаимодействия, обращаясь к системным средствам. Следует иметь в виду, что приложение может взять на себя функции некоторых верхних уровней модели *OSI*, в таком случае, при необходимости межсетевого обмена оно обращается напрямую к системным средствам, выполняющим функции оставшихся нижних уровней модели *OSI*. Так, в неразветвленных локальных сетях отпадает необходимость в средствах сетевого и транспортного уровней. В то же время сложность функций канального уровня делает целесообразным его разделение в локальных сетях на два подуровня: *управление доступом к каналу (MAC – Medium Access Control)* и *управление логическим каналом (LLC – Logical Link Control)*. К подуровню *LLC* в отличие от подуровня *MAC* относится часть функций канального уровня, не связанных с особенностями передающей среды.

Приложение конечного пользователя может использовать системные средства взаимодействия не только для организации диалога с другим приложением, выполняемым другой машиной, но и просто для получения услуг того или иного сетевого сервиса, например, доступа к удаленным файлам, получение почты или печати на разделяемом принтере.

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**, т. е. протоколы этих уровней тесно связаны с технической реализацией сети, с используемым коммуникационным оборудованием, и переход на другой тип оборудования означает полную смену протоколов физического и канального уровня во всех узлах сети.

Три верхних уровня – сеансовый, уровень представления и прикладной – ориентированы на приложения и мало зависят от

технических особенностей построения сети, т. е. **сетенезависимы**. На протоколы этих уровней не влияют никакие изменения в топологии сети, замена оборудования или переход на другую сетевую технологию.

Транспортный уровень является промежуточным, он скрывает все детали функционирования нижних уровней от верхних уровней. Это позволяет разрабатывать приложения, независимые от технических средств, непосредственно занимающихся транспортировкой сообщений.

Компьютер, с установленной сетевой ОС, взаимодействует с другим компьютером с помощью протоколов всех семи уровней. Это взаимодействие компьютеры осуществляют 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 и коммутатор), либо на физическом, канальном и сетевом, иногда захватывая и транспортный уровень (маршрутизатор).

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

Протоколы канального уровня не позволяют строить сети с развитой структурой, например, сети, объединяющие несколько сетей предприятия в единую сеть, или высоконадежные сети, в которых существуют избыточные связи между узлами. Для того чтобы, с одной стороны, сохранить простоту процедур передачи пакетов для типовых топологий, а с другой стороны, допустить использование произвольных топологий, вводится дополнительный сетевой уровень.

В протоколах сетевого уровня термин «сеть» означает совокупность компьютеров, соединенных между собой в соответствии с одной из стандартных типовых топологий и использующих для передачи пакетов общую базовую сетевую технологию. Внутри сети сегменты не разделяются

маршрутизаторами, иначе это была бы не одна сеть, а несколько сетей. Маршрутизатор соединяет несколько сетей в интернет.

Сети соединяются между собой специальными устройствами, маршрутизаторами, собирающими информацию о топологии межсетевых соединений и на ее основании пересылающие пакеты сетевого уровня в сеть назначения. Проблема выбора наилучшего пути называется маршрутизацией, и ее решение является главной задачей сетевого уровня. Критерием при выборе маршрута являются время или надежность передачи данных и др. Протоколы сетевого уровня реализуются программными модулями операционной системы, а также программными и аппаратными средствами маршрутизаторов.

Основная идея введения сетевого уровня заключается в сохранении в неизменном виде технологий, используемых в объединяемых сетях, но с добавлением в кадры канального уровня дополнительной информации – заголовка сетевого уровня, на основании которого можно было бы находить адресата в сети с любой базовой технологией. Заголовок пакета сетевого уровня имеет унифицированный формат, не зависящий от форматов кадров канального уровня тех сетей, которые могут входить в объединенную сеть.

Заголовок сетевого уровня должен содержать адрес назначения и другую информацию, необходимую для успешного перехода пакета из сети одного типа в сеть другого типа. К такой информации может относиться, например:

- номер фрагмента пакета, нужный для успешного проведения операций сборки-разборки фрагментов при соединении сетей с разными максимальными размерами кадров канального уровня;

- время жизни пакета, указывающее, как долго он путешествует по интернету, это время может использоваться для уничтожения «заблудившихся» пакетов;

- информация о наличии и о состоянии связей между сетями, помогающая узлам сети и маршрутизаторам рационально выбирать межсетевые маршруты;

- информация о загруженности сетей, также помогающая согласовать темп посылки пакетов в сеть конечными узлами с

реальными возможностями линий связи на пути следования пакетов;

- качество сервиса – критерий выбора маршрута при межсетевых передачах – например, узел-отправитель может потребовать передать пакет с максимальной надежностью, возможно в ущерб времени доставки.

В качестве адресов отправителя и получателя в составной сети используется не MAC-адрес, а пара чисел – номер сети и номер компьютера в данной сети. В канальных протоколах поле «номер сети» обычно отсутствует – предполагается, что все узлы принадлежат одной сети. Явная нумерация сетей позволяет протоколам сетевого уровня составлять точную карту межсетевых связей и выбирать рациональные маршруты при любой их топологии, используя альтернативные маршруты, если они имеются, что не делают мосты.

Таким образом, внутри сети доставка сообщений регулируется канальным уровнем. А вот доставкой пакетов между сетями занимается сетевой уровень.

Существует два подхода к назначению номера узла в заголовке сетевого пакета. Первый основан на использовании для каждого узла нового адреса, отличного от того, который использовался на канальном уровне. Преимуществом такого подхода является его универсальность и гибкость, каков бы ни был формат адреса на канальном уровне, формат адреса узла на сетевом уровне выбирается единым. Однако здесь имеются и некоторые неудобства, связанные с необходимостью заново нумеровать узлы, причем чаще всего вручную.

Второй подход состоит в использовании на сетевом уровне того же адреса узла, что был дан ему на канальном уровне. Это избавляет администратора от дополнительной работы по присвоению новых адресов, снимает необходимость в установлении соответствия между сетевым и канальным адресом одного и того же узла, но может породить сложную задачу интерпретации адреса узла при соединении сетей с разными форматами адресов.

4.3. Стандартизация сетей

Без согласования всеми производителями общепринятых стандартов для оборудования и протоколов прогресс в деле «строительства» сетей не был бы возможен. Поэтому все развитие компьютерной отрасли, отражено в стандартах – любая новая технология только тогда приобретает «законный» статус, когда ее содержание закрепляется в соответствующем стандарте.

В компьютерных сетях идеологической основой стандартизации является модель взаимодействия открытых систем (*OSI*).

Существует несколько организационных подразделений, отвечающих за развитие и за стандартизацию архитектуры и протоколов Интернета. Основным из них является научно-административное сообщество Интернета (*Internet Society, ISOC*), занимающееся социальными, политическими и техническими проблемами эволюции Интернета.

Под управлением *ISOC* работает совет по архитектуре Интернета (*Internet Architecture Board, IAB*). В *IAB* входят две основные группы: *Internet Research Task Force (IRTF)* и *Internet Engineering Task Force (IETF)*. *IRTF* координирует долгосрочные исследовательские проекты по протоколам TCP/IP. *IETF* – это инженерная группа, которая занимается решением текущих технических проблем Интернета. *IETF* определяет спецификации, которые затем становятся стандартами Интернета.

4.3.1. Стандартные стеки коммуникационных протоколов

Важнейшим направлением стандартизации в области локальных сетей является стандартизация коммуникационных протоколов. Наиболее известными стеками протоколов являются: OSI, TCP/IP, IPX/SPX, NetBIOS/SMB, DECnet, SNA (не все из них применяются сегодня на практике).

Все эти стеки, кроме SNA на нижних уровнях – физическом и канальном, используют одни и те же хорошо стандартизованные протоколы Ethernet, Token Ring, 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 рекомендуемой модели *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 *OSI*. В отличие от других стеков протоколов стек *OSI* полностью соответствует модели *OSI*, он включает спецификации протоколов для всех семи уровней взаимодействия, определенных в этой модели. На нижних уровнях стек *OSI* поддерживает Ethernet, Token Ring, FDDI, протоколы глобальных сетей, X.25 и ISDN, т. е. использует разработанные вне стека протоколы нижних уровней, как и все другие стеки. Протоколы сетевого, транспортного и сеансового уровней стека *OSI* специфицированы и реализованы различными производителями, но распространены пока мало. Наиболее популярными протоколами стека *OSI* являются прикладные протоколы. К ним относятся: протокол передачи файлов РТАМ, протокол эмуляции терминала VTP, протоколы справочной службы X.500, электронной почты X.400 и ряд других.

Протоколы стека *OSI* отличаются большей сложностью и неоднозначностью спецификаций. Эти свойства явились результатом общей политики разработчиков стека, стремившихся учесть в своих протоколах все случаи жизни и все существующие и появляющиеся технологии.

Из-за своей сложности протоколы *OSI* требуют больших затрат вычислительной мощности центрального процессора, что делает их наиболее подходящими для мощных машин, а не для сетей персональных компьютеров.

Стек *TCP/IP*. *Transmission Control Protocol/Internet Protocol (TCP/IP)* – это промышленный стандарт стека протоколов, разработанный для глобальных сетей. Разработан по инициативе Министерства обороны США более 20 лет назад для связи экспериментальной сети ARPAnet с другими спутниковыми сетями как набор общих протоколов для разнородной вычислительной среды. Сеть ARPA поддерживала

разработчиков и исследователей в военных областях. В сети ARPA связь между двумя компьютерами осуществлялась с использованием протокола *Internet Protocol (IP)*, который и по сей день является одним из основных в стеке TCP/IP и фигурирует в названии стека.

Большой вклад в развитие стека TCP/IP внес университет Беркли, реализовав протоколы стека в своей версии ОС UNIX. Широкое распространение ОС UNIX привело и к широкому распространению протокола IP и других протоколов стека. На этом же стеке работает всемирная информационная сеть Internet, чье подразделение *Internet Engineering Task Force (IETF)* вносит основной вклад в совершенствование стандартов стека, публикуемых в форме спецификаций RFC.

В настоящее время стек TCP/IP распространен в сетях с ОС UNIX, Windows NT, NetWare 4.1, Windows 2000, XP, VISTA, 2007, 2008, что является хорошей предпосылкой для быстрого роста числа установок стека TCP/IP.

Лидирующая роль стека TCP/IP объясняется следующими его свойствами:

- это наиболее завершенный стандартный и в то же время популярный стек сетевых протоколов, имеющий многолетнюю историю;
- почти все большие сети передают основную часть своего трафика с помощью протокола TCP/IP;
- это метод получения доступа к сети Internet;
- этот стек служит основой для создания intranet – корпоративной сети, использующей транспортные услуги Internet и гипертекстовую технологию WWW, разработанную в Internet;
- все современные операционные системы поддерживают стек TCP/IP;
- это гибкая технология для соединения разнородных систем, как на уровне транспортных подсистем, так и на уровне прикладных сервисов;
- это устойчивая масштабируемая межплатформенная среда для приложений клиент-сервер.

4.3.2. Сетезависимые и сетенезависимые уровни стека TCP/IP

Рассматривая многоуровневую архитектуру TCP/IP, можно выделить в ней, подобно архитектуре *OSI*, уровни, функции которых зависят от конкретной технической реализации сети, и уровни, функции которых ориентированы на работу с приложениями (рис. 4.2).

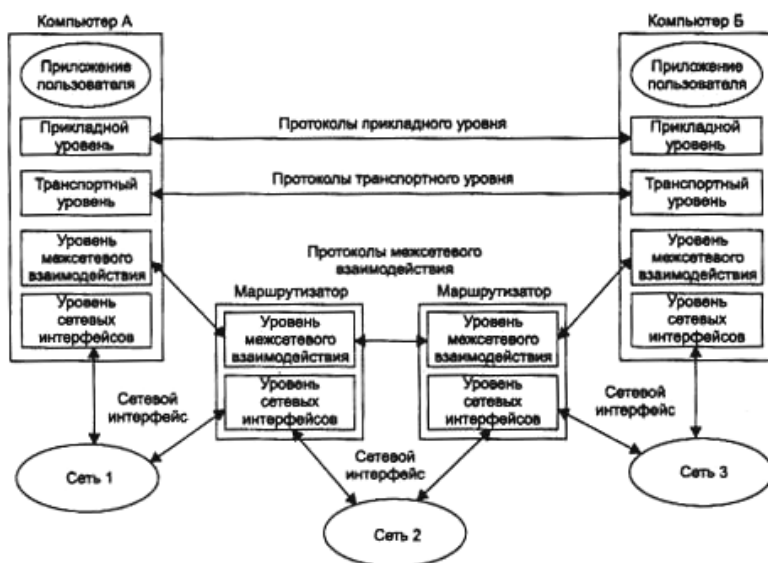


Рис.4.2. Сетезависимые и сетенезависимые уровни стека TCP/IP

Протоколы прикладного уровня стека TCP/IP работают на компьютерах, выполняющих приложения пользователей. Даже полная смена сетевого оборудования в общем случае не должна влиять на работу приложений, если они получают доступ к сетевым возможностям через протоколы прикладного уровня.

Протоколы транспортного уровня зависят от сети уже в большей степени, так как они реализуют интерфейс к уровням, непосредственно организующим передачу данных по сети. Однако подобно протоколам прикладного уровня, протоколы транспортного уровня устанавливаются только на конечных

узлах. Протоколы двух нижних уровней являются сетезависимыми, программные модули протоколов межсетевого уровня и уровня сетевых интерфейсов устанавливаются на всех хостах и шлюзах.

Каждый коммуникационный протокол оперирует с некоторой единицей передаваемых данных. Названия этих единиц иногда закрепляются стандартом, а чаще просто определяются традицией. В стеке TCP/IP за многие годы его существования образовалась устоявшаяся терминология в этой области (рис. 4.3).

Потоком (stream) называют данные, поступающие от приложений на вход протоколов транспортного уровня TCP и UDP.

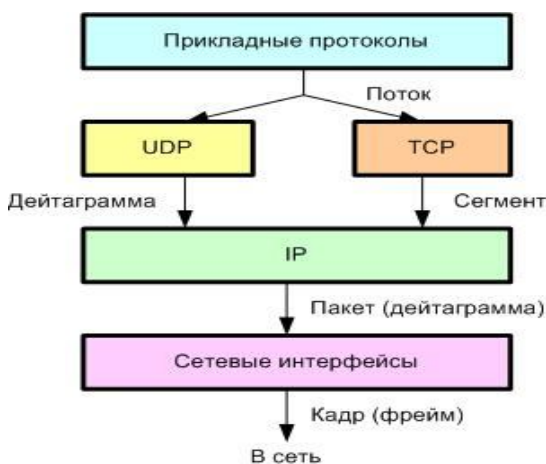


Рис.4.3. Название единиц данных, используемых в TCP/IP

Выводы:

Многоуровневая структура является эффективной моделью средств взаимодействия компьютеров в сети, в которой модули вышележащего уровня при решении своих задач рассматривают средства нижележащего уровня как некий инструмент. Каждый уровень структуры поддерживает: интерфейсы услуг с выше- и нижележащими уровнями «своей» иерархии средств; одноранговый интерфейс со средствами другой взаимодействующей стороны, расположенными на том же уровне иерархии.

Протоколы нижних уровней часто реализуются комбинацией программных и аппаратных средств, а протоколы верхних уровней – программными средствами.

Модель OSI содержит описание обобщенного представления средств сетевого взаимодействия и используется в качестве универсального языка сетевых специалистов, и называется справочной моделью. Модель OSI определяет 7 уровней взаимодействия: прикладной, представительный, сеансовый, транспортный, сетевой, канальный, физический, указывает, какие функции должен выполнять каждый уровень.

Стек TCP/IP используется для связи десятков миллионов компьютеров всемирной информационной сети Интернет. Стек TCP/IP распространен в сетях с ОС UNIX, Windows NT, NetWare 4.1, Windows 2000, XP, VISTA, 2007, 2008.

Контрольные вопросы:

1. Модель OSI.
2. Что стандартизирует модель OSI?
3. На каком уровне модели OSI работает прикладная программа?
4. На каком уровне модели OSI работают сетевые службы?
5. Модели сетевого взаимодействия.
6. Функции сетевого уровня.
7. Принципы взаимодействия информационных сетей.
8. Какая задача решается на физическом уровне?
9. На каком уровне реализуются механизмы обнаружения и коррекции ошибок?
10. Какие задачи решаются на канальном и сетевом уровнях?
11. Какой уровень отвечает за преобразование форматов данных?
12. На каком уровне выполняется шифрование и дешифрование данных?
13. Какие уровни модели OSI являются сетезависимыми?
14. Какие уровни модели OSI являются сетенезависимыми?
15. Какие свойства определяют лидирующую роль стека TCP/IP?

5. СЕТЕВАЯ ТЕХНОЛОГИЯ ETHERNET

Стандарт Ethernet был принят в 1984 г. Основным принципом, положенным 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.

Ethernet – базовая технология локальной сети, использующая метод множественного доступа к среде передачи с прослушиванием несущей и обнаружением коллизий (*CSMA/CD – Carrier Sense Multiple-Access/Collision Detection*). Топология – общая шина (рис. 2.3, б). Скорость передачи 10 Мб/с. Протокол стандартизован в спецификации IEEE 802.3. Стандарт разработан институтом инженеров по электронике и радиоэлектронике – (*Institute of Electrical and Electronics Engineers, IEEE*) – национальной организацией США, определяющей сетевые стандарты локальных сетей. В 1981 г. рабочая группа 802 сформулировала основные требования, которым должны удовлетворять локальные сети. Группа 802 определила множество стандартов, из них самыми известными являются стандарты: 802.1, 802.2, 802.3, 802.5, 802.11, описывающие общие понятия, используемые в области локальных сетей, а также стандарты на два нижних уровня сетей Ethernet, Token Ring, WLAN.

В стандарте Ethernet строго зафиксирована топология электрических связей. Компьютеры подключаются к разделяемой среде в соответствии с типовой структурой «общая шина». 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.

IEEE 802.3 и Ethernet совместимы друг с другом. Они являются простыми, устойчивыми и недорогими сетевыми

решениями. Узлы передают сообщения со скоростью 10 Мбит/с. В обоих случаях используется метод доступа CSMA/CD.

Различия между этими стандартами находятся в форматах кадров. Наиболее важные из них – различия между типами и длиной полей в кадрах. Эти различия иногда приводят к несовместимости этих стандартов.

5.1. Методы доступа

Различают случайные и детерминированные методы доступа к среде передачи данных.

5.1.1. Случайные методы доступа

При случайном методе доступа узлы сети выполняют передачу при наличии данных. Узлы используют канал без учета того, что делают остальные узлы сети. Если два и более узлов будут вести передачу пакетов одновременно, то происходит столкновение (*Collision*, *коллизия*, *конфликт*). В результате этого все передаваемые пакеты искажаются. Различают следующие случайные методы доступа: множественный доступ с обнаружением (проверкой) конфликтов; множественный доступ с проверкой несущей и т. д.

Проверка несущей (*Carrier Sense*) позволяет существенно уменьшить вероятность столкновений, и реализуется в сетевом адаптере. Узлы должны уметь определять коллизии (*Collision Detection*) как можно раньше, а не по окончании отправки кадра. Это экономит время и пропускную способность канала, как в методе CSMA/CD.

Множественный доступ с контролем несущей и обнаружением конфликтов (CSMA/CD - Carrier Sense Multiple Access with Collision Detection). Когда какой-то узел хочет передать пакет, он пытается вначале определить, что никакой другой узел в это время ничего не передает, т. е. среда передачи данных свободна. Операция проверки состояния канала называется проверкой несущей.

Если канал передачи занят, узел ждет пока канал не освободится. Если канал свободен, узел начинает передачу и во

время передачи продолжает «слушать» канал с целью выявления конфликта. В случае обнаружения конфликта, обе станции прекращают передачу. В сеть передается сигнал задержки (*jam pattern*), информирующий другие узлы сети о возникшем конфликте. Далее узлы ждут случайное время, а затем возобновляют попытку передачи данных. После 16 конфликтов сетевой адаптер прекращает попытку передачи данного кадра. Метод CSMA/CD широко используется в локальных сетях Ethernet DIX, Fast Ethernet и т. д.

Множественный доступ с контролем несущей и предотвращением конфликтов (CSMA/CA – Carrier Sense Multiple Access with Collision Avoidance). Этот метод – модифицированный вариант протокола CSMA/CD, разработанный для IEEE 802.11. Станция, которая собирается передавать информацию, сначала «слушает эфир». Если среда передачи свободна, то станция сначала ожидает в течение некоторого случайного промежутка времени, потом снова «слушает эфир» и, если среда передачи данных все еще свободна, осуществляет передачу. Если информация приходит без искажений (целостность данных проверяется по контрольной сумме), принимающая станция посылает обратно подтверждение. Если подтверждение не получено, то пакет передается снова через случайный промежуток времени.

При использовании режима принудительной очистки эфира *RTS/CTS* перед тем, как послать данные в эфир, станция сначала отправляет специальное сообщение – *RTS (Ready To Send)*. *RTS-сообщение* содержит информацию о продолжительности предстоящей передачи, об адресате. Это позволяет другим узлам задержать передачу на время, указанное в *RTS-сообщении*. Приемная станция, получив сигнал *RTS*, при готовности узла к приему информации, отвечает посылкой сообщения *CTS (Clear To Send)*.

5.1.2. Детерминированные методы доступа

При использовании этого метода любой узел может вычислить максимальное время, по истечении которого, он сможет начать передачу своих данных. Используется в таких

сетях как IEEE 802.5/Token Ring, FDDI и т. д. Среди детерминированных методов наиболее часто используются маркерные методы доступа.

Маркерный метод основан на передаче полномочий передающей станции с помощью специального пакета, называемого *маркером*. В компьютерных сетях широко применяется *метод передачи маркера по кольцу*.

В сетях IEEE 802.5/Token Ring применяется метод передачи маркера по кольцу. Здесь маркер – это трех байтовое сообщение, обладатель которого получает право на передачу данных в сеть. Если узел, принимающий маркер, не имеет данных для передачи, он просто переправляет маркер к следующему узлу. Каждый узел может удерживать маркер в течение определенного максимального времени (10 мс). Поэтому когда узел хочет передать данные, он должен дождаться свободного маркера. Когда кадр достигает адресата, то данные копируются, устанавливается флаг подтверждения приема, и кадр посылается дальше по кольцу. Когда кадр возвращается на передавший его узел, маркер и данные удаляются из кольца, а затем в кольцо передается новый маркер.

Доступ по приоритету запроса (метод *Demand Priority*) – это метод доступа, разработанный для сети со скоростью передачи 100 Мбит/с (100VG-AnyLAN). В этом методе концентратор выполняет функций арбитра, решающего проблему доступа к среде передачи данных. Узел сети, пожелавший передать пакет, посылает специальный сигнал концентратору, указывающий его приоритет. Концентраторы последовательно опрашивают каждый узел в сети и выявляют запросы на передачу.

Если сеть свободна, то концентратор разрешает передачу пакета. Если сеть занята, концентратор ставит полученный запрос в очередь. Запросы обрабатываются в соответствии с порядком их поступления и с учетом их приоритетов. Если одновременно принимаются два запроса, то первым обслуживается запрос с более высоким приоритетом. Если два запроса имеют одинаковый приоритет, то запросы обслуживаются в порядке очереди.

5.2. Форматы кадров технологии Ethernet

5.2.1. Ethernet DIX (II)

На рис. 5.1 приведен формат кадра по стандарту Ethernet DIX.

8	6	6	2	46-1500	4
Преамбула	АН	АО	T	Данные	КС

Рис. 5.1. Формат кадра Ethernet DIX

Ethernet DIX содержит следующие поля:

Поле «*Преамбула*» (*Preamble*). Это поле длиной 8 байт используется для синхронизации кадра. Первые семь байтов имеют вид 10101010, а последний байт – 0101011. Поле «*Адрес назначения DA*» (*Destination Address*), 6 байт. Содержит адрес узла сети, которому предназначено сообщение. Поле «*Адрес отправителя SA*» (*Source Address, адрес источника*), 6 байт. Содержит адрес узла-отправителя кадра. Поле «*Tun*» (*Type, T или EtherType*), 2 байта. Поле предназначено для указания типа протокола верхнего уровня, вложившего свой пакет в поле данных этого кадра. Например, протокол IP имеет код 2048₁₀ (0800₁₆). Поле «*Данных*» (*Data*) имеет длину от 46 до 1500 байт и содержит данные для передачи. Поле «*Контрольная сумма FCS*» (*Frame Check Sequence*), 4 байта. Значение этого поля вычисляется на основе содержимых заголовка (без учета преамбулы) и поля данных с помощью 32-тиразрядного циклического избыточного кода CRC-32. После получения кадра принимающий узел выполняет собственное вычисление контрольной суммы для этого кадра. Затем сравнивает полученное значение со значением поля контрольной суммы. В случае несовпадения пакет считается испорченным и игнорируется.

Длина кадра – от 64 до 1518 байт, не считая преамбулы и признака начала кадра. Для Ethernet DIX минимальная длина передаваемых данных составляет 64 байтов.

5.2.2. Стандарт IEEE 802.3

На рис. 5.2 приведен формат кадра по стандарту IEEE 802.3.

7	1	2/6	2/6	2	0-1500		4
Преамбула	ПНК	АН	АО	Длина	Данные	Заполнитель	КС

Рис. 5.2. Формат кадра IEEE 802.3

Поле «*Преамбула*» (*Preamble*) состоит из семи байт синхронизации (10101010). Поле «*Признак начала кадра SFD*» (*Start-of-frame-delimiter*) состоит из одного байта 10101011. Поле «*Адрес назначения DA*» (*Destination Address*) и Поле «*Адрес отправителя SA*» (*Source Address*) содержат адреса узлов получателя и отправителя кадра. Поле «*Длина L*» (*Length*). Определяет длину поля данных в кадре. Поле «*Данные*» (*Data*). Длина поля от 0 до 1500 байт. Если длина поля меньше 46 байт, то используется поле заполнения, чтобы дополнить кадр до минимально допустимого значения в 46 байт. Поле «*Заполнения*» (*Padding*) состоит из такого количества байт заполнителей, которое обеспечивает минимальную длину поля данных в 46 байт. Поле «*Контрольная сумма FCS*» (*Frame Check Sequence*). Это значение вычисляется также по алгоритму CRC-32. Используется для выявления ошибок передачи.

Кадр 802.3 является кадром MAC-подуровня, поэтому в соответствии со стандартом 802.2 в его поле данных вкладывается кадр подуровня LLC с удаленными флагами начала и конца кадра.

После окончания передачи кадра все узлы сети обязаны выдержать технологическую паузу *IPG* (*Inter Packet Gap*) в 9,6 мкс. Эта пауза называется также *межкадровым интервалом*. Пауза нужна для приведения сетевых адаптеров в исходное состояние, а также для предотвращения монопольного захвата передающей среды одним узлом. После окончания технологической паузы узлы имеют право начать передачу своего кадра.

5.2.3. Fast Ethernet (IEEE 802.3u)

Технология Fast Ethernet является развитием классической технологии Ethernet (10 Мбит/с). Ее основные характеристики:

- увеличение пропускной способности сегментов сети до 100 Мбит/с;
- Fast Ethernet также использует метод случайного доступа CSMA/CD;
- сохранение звездообразной топологии сетей;
- возможность работы на всех основных типах кабелей, т.е. поддержка витой пары и оптоволоконного кабеля;
- большая преемственность по отношению к классическому Ethernet, т. е. Fast Ethernet не требует коренного переобучения персонала и замены оборудования во всех узлах сети.

В стандарте Fast Ethernet подуровни LLC и MAC не изменились. Форматы кадров технологии Fast Ethernet не отличаются от форматов кадров технологий 10-мегабитного Ethernet.

Все временные параметры передачи кадров Fast Ethernet в 10 раз меньше соответствующих времен технологии 10-Мегабитного Ethernet: битовый интервал составляет 10 нс вместо 100 нс, а межкадровый интервал (IPG) – 0.96 мкс вместо 9.6 мкс соответственно.

Все временные параметры алгоритма доступа (интервал отсрочки, время передачи кадра минимальной длины и т. п.), измеренные в битовых интервалах, остались прежними, поэтому в MAC-подуровне, изменений нет.

Используются следующие служебные символы:

- *Idle, I* (простой) – 11111;
- *Start Delimiter 1* (первый символ границы начала кадра) – 11000 (*J*);
- *Start Delimiter 2* (второй символ границы начала кадра) – 10001 (*K*);
- *Ending Delimiter* (конец кадра) – 01101 (*T*).

Отличия Fast Ethernet от классического Ethernet сосредоточены на физическом уровне.

Для технологии Fast Ethernet разработаны различные варианты физического уровня, отличающиеся не только типом кабеля и электрическими параметрами импульсов, но и

способом кодирования сигналов и количеством используемых в кабеле проводников. Поэтому физический уровень Fast Ethernet имеет более сложную структуру, чем классический Ethernet. Физический уровень состоит из элементов:

- уровень согласования (*reconciliation sublayer*);
- независимый от среды интерфейс (*Media Independent Interface, MII*);
- устройство физического уровня (*Physical layer device, PHY*).

Подуровень согласования нужен для того, чтобы подуровень MAC (рассчитанный на интерфейс AUI) мог работать с физическим уровнем через интерфейс MII.

Интерфейс MII поддерживает независимый от используемой физической среды способ обмена данными между MAC-подуровнем и подуровнем PHY.

Устройство физического уровня (PHY) обеспечивает кодирование данных, поступающих от MAC-подуровня для передачи их по кабелю определенного типа, синхронизацию передаваемых по кабелю данных, а также прием и декодирование данных в узле-приемнике. Состоит из следующих подуровней:

- подуровня логического кодирования данных (PCS), преобразующего поступающие от подуровня MAC байты в символы кода 4B/5B или 8B/6T;
- подуровня физического присоединения (PMA) и подуровня зависимости от физической среды (PMD), которые обеспечивают формирование сигналов в соответствии с методом физического кодирования (NRZI или MLT-3);
- подуровня «автопереговоров», который позволяет двум взаимодействующим портам автоматически выбрать наиболее эффективный режим работы (полудуплексный или полнодуплексный).

Различают 3 спецификации для поддержки следующих типов кабелей:

- *100Base-TX* для двухпарного кабеля на витой паре UTP-5 или STP Type 1;
- *100Base-T4* для четырехпарного кабеля UTP категории 3, 4 или 5 и STP Типа 1;
- *100Base-FX* для многомодового оптоволоконного кабеля.

100Base-FX (многомодовое оптоволокно). Эта спецификация определяет работу Fast Ethernet по многомодовому оптоволокну в полудуплексном и полнодуплексном режимах. В стандарте Fast Ethernet определен метод логического кодирования 4В/5В. При этом методе каждые 4 бита данных MAC-подуровня представляются 5 битами. Для представления 5-тибитовых порций в виде оптических или электрических сигналов используется метод физического кодирования NRZI. Стандарт 100Base-FX ориентирован в основном на магистрали – на соединение повторителей Fast Ethernet в пределах одного здания.

100Base-TX (двухпарная витая пара). Стандарт 100Base-TX требует применения двух пар UTP или STP. Одна пара служит для передачи, другая – для приема. Этим требованиям отвечают два основных кабельных стандарта: UTP Категории 5 и STP Типа 1. Спецификация 100Base-TX обеспечивает полнодуплексный режим работы. Используются метод логического кодирования 4В/5В и метод физического кодирования MLT-3 для передачи сигналов 5-битовых порций по витой паре, а также функция «автопереговоров» (*Auto-negotiation*) для выбора режима работы порта.

100Base-T4 (четырепарная витая пара). Разработана для того, чтобы использовать возможности витой пары категории 3. В 100Base-T4 используются все четыре пары кабеля UTP: одна для передачи, другая для приема, а оставшиеся две работают как на передачу, так и на прием. Таким образом, и прием, и передача данных могут осуществляться по трем парам, т. е. для 100Base-T4 нужны все четыре пары и поэтому полнодуплексный режим этим протоколом не поддерживается. Стандарт 100base-T4 предполагает применение метода физического кодирования MLT-3 и схемы кодирования 8В/6Т.

5.2.4. Высокоскоростная технология Gigabit Ethernet

Стандарт 802.3z был принят в 1998 году. Скорость передачи данных 1000 Мбит/с. В технологии Gigabit Ethernet по сравнению с технологиями Ethernet и Fast Ethernet:

- сохраняются все форматы кадров Ethernet;

- существуют полудуплексная версия протокола, поддерживающая метод доступа CSMA/CD, и полнодуплексная версия, работающая с коммутаторами;

- поддерживаются все основные виды кабелей, используемых в Ethernet и Fast Ethernet: волоконно-оптический, витая пара категории 5, коаксиал.

Для сохранения приведенных выше свойств внесены изменения не только в физический уровень, но и в уровень MAC.

Для расширения максимального диаметра сети Gigabit Ethernet в полудуплексном режиме до 200 м увеличен минимальный размер кадра (без учета преамбулы) с 64 до 512 байт. Для увеличения длины кадра до требуемой величины используется *поле расширения (extention)*. Это поле заполняется запрещенными символами кода 8B/10B, которые невозможно принять за коды данных. Формально минимальный размер кадра не изменился, он по-прежнему равен 64 байтам, т. к. поле расширения помещается после контрольной суммы кадра FCS, его значение не включается в контрольную сумму и не учитывается значением поля L/T.

Для сокращения накладных расходов при передаче коротких кадров разрешено конечным узлам передавать несколько кадров подряд. Такой режим называется *Burst Mode – монопольный пакетный режим*. Станция может передать подряд несколько кадров с общей длиной не более 8192 байт. В полнодуплексном режиме поля *Extention* и *Burst Mode* не используются.

В стандарте 802.3z определены следующие типы физической среды:

- одномодовый волоконно-оптический кабель;
- многомодовый волоконно-оптический кабель 62,5/125;
- многомодовый волоконно-оптический кабель 50/125;
- двойной коаксиал с волновым сопротивлением 75 Ом.

Для кодирования сигналов, передающихся по оптоволокну, применяется схема 8B/10B.

В *1000Base-SX* используется длина волны 850 нм (*Short Wavelength, короткая волна*). Для *1000Base-SX* предельная длина оптоволоконного сегмента для кабеля 62,5/125 оставляет

220 м, а для кабеля 50/125 – 500 м. Можно увеличить длину кабеля до 800 м.

В *1000Base-LX* используется длина волны 1300 нм (*Long Wavelength, длинная волна*). В качестве источника излучения применяется полупроводниковый лазерный диод. В основном применяется одномодовое оптоволокно. Максимальная длина кабеля для одномодового волокна равна 5000 м. *1000Base-LX* может работать и на многомодовом кабеле. В этом случае предельное расстояние – 550 м.

1000Base-CX использует короткий экранированный 150-тиомный медный кабель. В качестве среды передачи данных используется высококачественный твинаксиальный кабель (*Twinax*) с волновым сопротивлением 150 Ом (2x75 Ом). В сегменте 2 узла. Максимальная длина твинаксиального сегмента составляет всего 25 метров. При этом получается режим полудуплексной передачи. Для обеспечения полнодуплексной передачи необходимы еще две пары коаксиальных проводников. При этом используется специальный кабель (*Quad-кабель*), который содержит четыре коаксиальных проводника.

1000Base-T. Стандарт (IEEE 802.3ab) физического уровня *1000Base-T* для четырехпарных кабелей UTP5 длиной 100 м (т. е. для сетей с диаметром 200 м). В сегменте 2 узла. Для кодирования данных применяется код PAM5, использующий 5 уровней потенциала. Четыре из них используется для кодирования информационных битов (00, 01, 10, 11), а пятый предназначен для исправления ошибок.

Выводы:

В технологии Ethernet на разделяемой среде применяется случайный метод доступа CSMA/CD, ему свойственна коллизия, являющаяся следствием этого метода.

Коллизия – ситуация, когда две станции одновременно пытаются передать кадр данных через общую среду.

В зависимости от типа физической среды стандарт IEEE 802.3 определяет различные спецификации Ethernet со скоростью 10 Мбит/с: 10Base-5, 10Base-2, 10Base-T, FOIRL, 10Base-FL, 10Base-FB.

Проверка несущей позволяет существенно уменьшить вероятность столкновений, и реализуется в сетевом адаптере.

Маркерный метод основан на передаче полномочий передающей станции с помощью специального пакета, называемого маркером. В компьютерных сетях широко применяется метод передачи маркера по кольцу.

В сетях Token Ring используется детерминированный метод доступа с передачей токена. Логической топологией сетей Token Ring является кольцо, физической – звезда. За счет кольцевой топологии технология Token Ring отчасти обеспечивает отказоустойчивость.

Контрольные вопросы:

1. Почему протоколы канального уровня технологий глобальных сетей не делятся на подуровни MAC и LLC?
2. Какие функции выполняет уровень LLC?
3. Чем объясняется, минимальный размер поля данных кадра Ethernet в 46 байт?
4. Какое максимальное время ожидания выделено адаптером, прежде чем кадр будет отброшен из-за постоянных коллизий при передаче?
5. Объясните понятие «коллизия».
6. Какие используются методы передачи данных на канальном уровне?
7. Как называется операция проверки состояния канала?
8. В чем различие между случайными и детерминированными методами доступа?
9. Какую функцию выполняет в сети сигнал задержки?
10. Какую информацию содержит RTS-сообщение?
11. Какой алгоритм применяется при определении контрольной суммы в поле Frame Check Sequence кадра IEEE 802.3?
12. Какие основные характеристики технологии Fast Ethernet?
13. Какой метод логического кодирования определен в стандарте Fast Ethernet?
14. Какой метод физического кодирования используется в стандарте 100Base-TX?
15. Какая применяется схема кодирования сигналов, передающихся по оптоволокну?

6. БЕСПРОВОДНЫЕ СЕТИ

Стандарт 802.11 – это стандарт беспроводных сетей (*WLAN* – *Wireless/Wale Local Area Network*), в котором используется метод множественного доступа с прослушиванием несущей и исключением коллизий (*CSMA/CD* – *Carrier Sense Multiple Access with Collision Avoidance*). Этот метод требует от станции отправителя положительного подтверждения (*ACK*) каждого принятого кадра.

По радиусу действия и назначению современные беспроводные сети подразделяются на локальные *WLAN* (*Wireless Local Area Network*), персональные *WPAN* (*Wireless Personal Area Network*), городские *WMAN* (*Wireless Metropolitan Area Network*,) и глобальные *WWAN* (*Wireless Wide Area Network*).

6.1. Беспроводные локальные сети

Недостатками беспроводных сетей являются: низкая скорость передачи данных, нестабильность соединения, малая защищенность при передаче конфиденциальной информации, легкость их перехвата, возможность препятствий на пути радиоволн, увеличение вероятности ошибок при приеме. Еще одна проблема – «свободный статус» диапазона 2,4 ГГц. В этом диапазоне частот могут работать, например, микроволновые печи или медицинские приборы. Медицинский прибор может нарушить работу локальной сети, но хуже если сеть создаст серьезные помехи медицинскому прибору. Конечно, все перечисленные недостатки беспроводных сетей носят временный характер, но беспроводное решение уместнее рассматривать не как альтернативу Ethernet, а как дополнение, расширяющее возможности классических проводных сетей.

Существуют различные типы беспроводных сетей, отличающиеся друг от друга и радиусом действия, и поддерживаемыми скоростями соединения, и технологией кодирования данных. Можно отметить следующие спецификации семейства стандартов IEEE 802.11: IEEE 802.11 (первоначальный стандарт, 1997 г.), 11a (2001 г.), 11b (1999 г.),

11d (2001 г.), 11g (2003 г.), 11e (2005 г.), 11h (2004 г.), 11i (2004 г.), 11j (2004 г.), 11n (MIMO-технология, 2009 г.), 11s (Mesh-технология) и др.

Стандарт IEEE 802.11 является базовым стандартом и определяет протоколы, необходимые для организации беспроводных локальных сетей (WLAN). Основные из них – протокол управления доступом к среде MAC и протокол РНУ передачи сигналов в физической среде. В качестве последней допускается использование радиоволн и инфракрасного излучения. Стандарты IEEE802.11 работают на частотах 2,4 – 2,4835 ГГц, 5,15 – 5,35 ГГц и 5,725 – 5,825 ГГц, мощность передатчика в пределах 0 мВт – 1Вт, пропускная способность 11 Мбит/с, 54 Мбит/с и выше.

В основу стандарта 802.11 положена сотовая архитектура, причем сеть может состоять как из одной, так и нескольких ячеек. Каждая сота управляется базовой станцией, называемой *точкой доступа AP (Access Point)*, которая вместе с находящимися в пределах радиуса ее действия рабочими станциями пользователей, образует *базовую зону обслуживания BSS (Basic Service Set)*. Точки доступа многосотовой сети взаимодействуют между собой через распределительную систему *DS (Distribution System)*, представляющую собой эквивалент магистрального сегмента проводных локальных сетей. Вся инфраструктура, включающая точки доступа и распределительную систему образует *расширенную зону обслуживания ESS (Extended Service Set)*. Стандартом предусмотрен также односотовый вариант беспроводной сети, который может быть реализован и без точки доступа, при этом часть ее функций выполняются непосредственно рабочими станциями.

Стандарт IEEE 802.11 определяет два режима работы сети – режим «*Ad-hoc*» и *клиент/сервер* (или режим *инфраструктуры – infrastructure mode*).

Режим «*Ad-hoc*» (*точка-точка*, или независимый базовый набор служб, IBSS) – простая сеть, в которой связь между многочисленными станциями устанавливается напрямую, без использования специальной точки доступа. Для этого режима требуется, оснащение беспроводным адаптером каждой

станции. Основным недостатком режима «Ad-hoc» является ограниченный диапазон действия возможной сети и невозможность подключения к внешней сети (например, к Интернету).

В режиме *клиент/сервер* или *Infrastructure Mode* станции взаимодействуют друг с другом не напрямую, а через точку доступа *AP*, которая выполняет в беспроводной сети роль своеобразного концентратора. Рассматривают два режима взаимодействия с точками доступа – *BSS* и *ESS*. В режиме *BSS* все станции связываются между собой только через точку доступа, которая может выполнять также роль моста к внешней сети. В расширенном режиме *ESS* существует инфраструктура нескольких сетей *BSS*, причём сами точки доступа взаимодействуют друг с другом, что позволяет передавать трафик от одной *BSS* к другой. Сами точки доступа соединяются между собой с помощью либо сегментов кабельной сети, либо радиомостов.

Подобно проводной сети Ethernet, в беспроводных компьютерных сетях Wi-Fi канальный уровень включает в себя подуровни *LLC (Logical Link Control)* и *MAC (Media Access Control)*. У Ethernet и IEEE 802.11 один и тот же *LLC*, что значительно упрощает объединение проводных и беспроводных сетей. *MAC* у обоих стандартов имеет много общего, однако есть некоторые различия.

Стандарт 802.11 определяет три класса кадров, передаваемых по каналу: *информационные, служебные и управляющие*. Формат *информационного кадра* показан на рис. 6.1. Заголовок кадра состоит из 7 полей с общей длиной 30 байт.

2 байт	2	6	6	6	2	6	0-2312	4
Управление кадром	Длительность	Адрес 1	Адрес 2	Адрес 3	Номер	Адрес 4	Данные	Контрольная сумма

Рис. 6.1. Информационный кадр стандарта IEEE 802.11

Поле «Управление кадром» (*Frame Control*) имеет длину 2 байта (рис. 6.2). В поле «Длительность» задается время, которое будет потрачено на передачу кадра и подтверждения. Имеется четыре адреса в формате, соответствующем стандарту IEEE 802. Первые два адреса – это адреса отправителя и получателя. Кадры могут входить в ячейку или покидать ее через базовую станцию. Поэтому следующие два адреса хранят адреса исходной и целевой ячеек при передаче трафика между ячейками. Поле «Номер» позволяет нумеровать фрагменты. Поле «Данные» содержит передаваемую информацию (0 – 2312 байт). Поле «Контрольная сумма» предназначено для проверки целостности принятого кадра. Его значение подсчитывается с помощью алгоритма CRC-32.

2 байт	2	4	1	1	1	1	1	1	1	1
Версия	Тип	Подтип	K DS	От DS	MF	Повтор	Питание	Продолжение	W	O

Рис. 6.2. Поле управления кадром стандарта IEEE 802.11

Поле «Управление кадром» содержит 11 полей. Поле «Версия протокола» позволяет двум протоколам работать одновременно в одной ячейке сети. Поле «Тип» указывает на информационный, служебный или управляющий. Поле «Подтип» указывает на RTS, CTS или ACK кадры. Биты «K DS» и «От DS» говорят о направлении движения кадра: к межсетевой системе распределения DS (например, Ethernet) или от нее. Поле «MF» (*More Fragments*) – бит промежуточного фрагмента. «MF=1» означает, что далее следует еще один фрагмент, а «MF=0» – последний или единственный фрагмент в сообщении. Бит «Повтор» маркирует повторно посылаемый фрагмент. Бит «Управление питанием» используется базовой станцией для переключения станции в режим пониженного потребления или выхода из этого режима. Бит «Продолжение» показывает, что у отправителя имеются еще кадры для пересылки. Бит «W=1» означает, что для шифрования в кадре использован алгоритм WEP. Бит «O=1» означает, что кадры с

этим битом должны обрабатываться в приемнике строго по порядку.

Управляющие кадры имеют формат, сходный с форматом информационных кадров, за одним исключением: в управляющем кадре отсутствуют поля базовых станций. Служебные кадры содержат один или два адреса, отсутствуют поля «Данные» и «Номер».

Физический уровень предусматривает передачу сигнала одним из следующих методов: *FHSS (Frequency Hopping Spread Spectrum)* – метод частотных скачков (скачкообразной перестройки частоты); *DSSS (Direct Sequence Spread Spectrum)* – метод прямой последовательности (широкополосная модуляция с прямым расширением спектра), *CCK (Complementary Code Keying)* – комплементарные коды, *OFDM (Orthogonal Frequency Division Multiplexing)* – метод ортогонального частотного разделения с мультиплексированием, *PBCC (Packet Binary Convolutional Coding)* – метод двоичного пакетного сверточного кодирования, технология гибридного кодирования (например, *CCK-OFDM*).

Существуют следующие типы модуляции: двоичная фазовая модуляция *BPSK (Binary Phase Shift Key)*, квадратурная фазовая модуляция *QPSK (Quadrature Phase Shift Key)*, относительная фазовая модуляция *DPSK (Differential Phase Shift Keying)*, относительная двоичная фазовая модуляция *DBPSK (Differential Binary Phase Shift Key)*, квадратурная относительная фазовая модуляция *DQPSK (Differential Quadrature Phase Shift Key)*, квадратурная амплитудная модуляция *QAM (Called Quadrature Amplitude Modulation)* – *16-QAM* и *64-QAM*.

Первый международный стандарт на беспроводные сети Wireless LAN разрабатывался с 1990 по 1997 гг. Однако 2 Мбит/с уже не удовлетворяли пользователей. В 1999 г. было принято расширение стандарта IEEE 802.11, получившее название IEEE 802.11b. Более высокоскоростная версия 802.11b или **Wi-Fi** (*Wireless Fidelity* – дословно переводится как «беспроводная точность») определяет скорость передачи данных до 11 Мбит/с. Она является прямым продолжением стандарта 802.11. Стандартом IEEE 802.11b предусмотрено использование частотного диапазона от 2,4 до 2,4835 ГГц. Без

подключения дополнительной антенны устойчивая связь для IEEE 802.11b достигается в среднем на расстояниях: открытое пространство – 500 м, комната, разделенная перегородками из неметаллического материала – 100 м, офис из нескольких комнат – 30 м.

Стандарт 802.11a, утвержденный в 1999 г., предусматривает передачу данных со скоростью до 54 Мбит/сек в частотном диапазоне 5 ГГц (от 5,15 до 5,350 ГГц и от 5,725 до 5,825 ГГц). К недостаткам 802.11a относятся более высокая потребляемая мощность радиопередатчиков для частот 5 ГГц, а также меньший радиус действия.

Стандарт 802.11g, принятый в июне 2003 г., имеет максимальную скорость передачи и составляет 54 Мбит/с. IEEE 802.11g является логическим развитием стандарта 802.11b и предполагает передачу данных в том же частотном диапазоне от 2,4 до 2,4835 ГГц. Если средний радиус сети стандарта IEEE 802.11a составляет 50 м, то радиус действия сетей 802.11g – порядка 100 м.

Стандарт 802.11i принят в июле 2004 г. Этот документ заменил существовавшее описание протокола WEP. Основное отличие официальной версии стандарта 802.11i от более раннего упрощенного варианта, опубликованного под названием WPA (сохранение алгоритма шифрования RC4) – применение AES шифрования в последнем варианте (WPA2).

Стандарт 802.11s. Один из способов увеличения радиуса действия беспроводной сети заключается в создании распределённой сети с использованием *Mesh-технологии* на основе стандарта IEEE 802.11s. Точки доступа *Mesh-сетей* (*Wireless Mesh Networks*, ячеистых сетей), кроме предоставления услуги абонентского доступа, выполняют ещё и функции маршрутизаторов/ретрансляторов для других точек доступа этой же сети. *Mesh-сети* – это перспективный класс беспроводных сетей, который широко применяется при построении локальных и городских беспроводных сетей.

Стандарт 802.11n в 2009 г. официально признан новым стандартом для сетей WiFi. Стандарт стал основой *MIMO (Multiple Input Multiple Output)* технологии, в которой сигнал передается и принимается несколькими антеннами одновременно. Устройства,

поддерживающие стандарт IEEE 802.11n, могут работать в частотном диапазоне либо 5, либо 2,4 ГГц. Пиковая скорость передачи колеблется от 300 до 450 Мбит/с.

6.2. Беспроводные персональные сети

Беспроводные персональные сети *WPAN (Wireless Personal Area Network)* служат для связывания между собой компонентов компьютера в пределах малого радиуса действия (в так называемой персональной зоне). Однако WPAN-сети нужны не только для подключения компьютерной периферии. Их работа основана на семействе стандартов IEEE 802.15. В эту группу входят стандарты IEEE 802.15.1, 802.15.2, 802.15.3, 802.15.4 и т. д.

6.2.1. Стандарт IEEE 802.15.1 (Bluetooth)

Bluetooth обеспечивает обмен информацией между такими устройствами: как карманные и обычные персональные компьютеры, мобильные телефоны, ноутбуки, принтеры, мышки, клавиатуры, джойстики, телефонные гарнитуры и т. д. Bluetooth позволяет этим устройствам обмениваться информацией без кабельного подключения в радиусе до 10–100 м друг от друга. Технология Bluetooth (стандарт IEEE 802.15.1 предназначена для построения персональных беспроводных сетей WPAN). Для организации компьютерных сетей ее возможностей недостаточно.

Изначально дальность действия радиointерфейса закладывалась равной 10 м, однако сейчас спецификациями Bluetooth уже она определена около 100 м. Для работы радиointерфейса Bluetooth используется частота 2,45 ГГц. Радиоканал обладает полной пропускной способностью в 1 Мбит/с, что обеспечивает создание асимметричного канала передачи данных на скоростях 723,3/57,6 Кбит/с или полнодуплексного канала на скорости 433,9 Кбит/с.

Спецификация Bluetooth, разработанная группой *Bluetooth Special Interest Group (Bluetooth SIG)*, стала частью стандарта IEEE 802.15.1 (2002 г.)

Bluetooth версии 2.0 (2004 г.) полностью совместим с версиями 1.x. Скорость передачи увеличена до 2,1 Мбит/с.

В Bluetooth версии 2.1 (2007 г.) добавлена энергосберегающая технология, позволяющая увеличить продолжительность работы устройства от одного заряда аккумулятора в 3–10 раз.

Формат кадра Bluetooth показан на рис. 6.3.

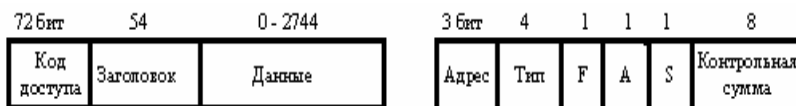


Рис. 6.3. Структура кадра Bluetooth и его заголовки

Поле «Код доступа» (72 бит) служит идентификатором главного узла. Это позволяет двум главным узлам, расположенным достаточно близко, «слышать» друг друга, различать, кому из них предназначаются данные. Поле «Заголовок» (54 бит), в котором 18-битный заголовок повторяется три раза. Поле «Данные» (0–2744 бит). Существуют кадры с длиной 1, 3 или 5 тактов. Поле «Данные» кадра с пятью тактами имеет размер 2744 бита.

Все кадры передаются между узлами по логическому каналу, называемый *соединением*. Существует два типа соединений. Первый называется *ACL* (*Asynchronous Connectionless* – асинхронный без установления связи). Он используется для коммутации пакетов данных, появляющихся в произвольный момент времени. У подчиненного узла может быть только одно *ACL-соединение* со своим главным узлом. Второй вид соединения называется *SCO* (*Synchronous Connection Oriented* – синхронный с установлением связи). Он предназначен для передачи данных в реальном масштабе времени – это требуется, например, при телефонных разговорах. У подчиненного узла может быть до трех соединений типа *SCO* с главным узлом.

Заголовок кадра состоит трех 18-битных заголовков, что в итоге составляет 54 бита. На принимающей стороне

анализируются все три копии каждого бита. Если они совпадают, бит принимается таким, какой он есть. В противном случае все решает большинство.

Поле «Адрес» идентифицирует устройство, которому предназначена информация. Поле «Тип» определяет тип передаваемого кадра (*ACL*, *SCO*, *опрос или пустой кадр*), метод коррекции ошибок и количество временных интервалов, из которых состоит кадр. Бит «F» (*Flow, поток*) выставляется подчиненным узлом и сообщает о наполненности буфера. Бит «A» (*Acknowledgement, подтверждение*) представляет собой подтверждение (*ACK*), отсылаемое заодно с кадром. Бит «S» (*Sequence, последовательность*) используется для нумерации кадров, позволяющий обнаруживать повторные передачи. Поле «Контрольная сумма заголовка» (8 бит) предназначена для контроля целостности заголовка кадра.

6.2.2. Стандарт IEEE 802.15.3

Стандарт IEEE 802.15.3 прямой наследник Bluetooth (частота 2,4 ГГц) и описывает работу малой беспроводной сети передачи данных – *пикосети (piconet)*. Пикосеть в стандарте IEEE 802.15.3 – это так называемая *ad hoc*-система, в которой несколько независимых устройств могут непосредственно взаимодействовать друг с другом. Размеры пикосети, как правило, не превышают 10 м. Основные требования к ней – высокая скорость передачи данных, простая инфраструктура, легкость установления соединения и вхождения в сеть, средства защиты данных и предоставление для определенных типов данных гарантированных параметров передачи (гарантия качества обслуживания, QoS).

Пикосеть может объединять несколько устройств, одно из которых выполняет функции управления (*piconet coordinator – PNC*). Стандарт также предусматривает возможность формирования так называемых дочерних пикосетей и описывает взаимодействие между независимыми соседними пикосетями.

В пикосети возможен обмен как *асинхронными*, так и *изохронными* (потокowymi) данными, например, звук и видео. Весь информационный обмен в пикосети основан на

последовательности суперкадров. Каждый кадр включает управляющий сегмент (*beacon*), интервал конкурентного доступа (*contention access period – CAP*) и набор временных интервалов (каналов), назначенных определенным устройствам. *PCN* определяет границы всех интервалов и распределяет каналы между устройствами. Во время *CAP* доступ к каналу происходит на основе механизма контроля несущей с предотвращением коллизий – CSMA/CA (как и в стандарте IEEE 802.11). То есть, кто первый успел занять канал, тот и работает. В этот период передаются команды или асинхронные данные.

Канальные интервалы (*CTA*) *PCN* назначает каждому устройству или группе устройств по запросу с их стороны. В управляющем сегменте для каждого из них задается момент начала и длительность. Назначение *CTA* для какого-либо устройства означает, что никакое другое устройство в этот момент не может работать на передачу. *CTA* могут динамически распределяться в суперкадре (для асинхронных и изохронных данных) или быть фиксированными (только для изохронных данных).

Спецификация физического канала в документе IEEE 802.15.3 приведена только для диапазона 2400–2483,5 МГц. Она предусматривает пять допустимых скоростей передачи: 11, 22, 33, 44 и 55 Мб/с. Скорость 22 Мбит/с является базовой, ее обязаны поддерживать все устройства IEEE 802.15.3. При работе на этой скорости данные не кодируются. В остальных случаях данные перед формированием модуляционных символов кодируются посредством сверточного кодера с 3-разрядным сдвиговым регистром (модуляция посредством решетчатого кода с 8 состояниями). При этом в кодере к исходному набору из 1/3/4/5 бит (при QPSK/16-QAM/32-QAM/64-QAM) добавляется кодовый бит с выхода трехразрядного сдвигового регистра.

Стандарт IEEE 802.15.3 требует, чтобы устройства могли работать в любом из пяти возможных частотных каналов: 2,412–2,462 МГц. Причем предусматривается два канальных плана – режим высокой плотности (четыре канала в допустимом диапазоне), и режим совместимости с сетью стандарта IEEE 802.11b (три разрешенных канала). Это означает, что каждое

устройство перед началом работы сканирует диапазон, находит свободные каналы, определяет наличие работающей сети 802.11b. Одновременно работать в такой сети могут до 250 пользователей. При возникновении помех со стороны других бытовых устройств или иных сетей, сети на основе IEEE 802.15.3 будут автоматически переключать каналы. Шифрование данных в сетях IEEE 802.15.3 может осуществляться по стандарту AES 128.

6.2.3. Стандарт IEEE 802.15.4 (ZigBee)

Разработчиком стандарта выступил альянс компаний (*Invensys, Honeywell, Mitsubishi Electric, Motorola, Philips и др.*), назвавший себя *ZigBee* (от *Zig-zag* – зигзаг и *Bee* – пчела). Подразумевалось, что топология сети будет напоминать зигзагообразную траекторию полета пчелы от цветка к цветку. Под таким замысловатым названием технология ZigBee и получает все большее распространение.

Стандарт IEEE 802.15.4 ориентирован на использование в качестве средства связи между автономными приборами и оборудованием. В корпоративном секторе это могут быть, например, складские системы, системы автоматизации производства, различные датчики, сенсоры, электронные метки. В домашних условиях – персональные компьютеры, игровые приставки, системы освещения и кондиционирования, радиофицированные игрушки и пульты дистанционного управления.

Стандарт IEEE 802.15.4 (ZigBee) предусматривает работу в трех диапазонах: один канал 868,0–868,6 МГц (для Европы); 10 каналов в диапазоне 902–928 МГц (шаг центральных частот – 2 МГц, самая нижняя из них – 906 МГц); 16 каналов в диапазоне 2450 МГц (шаг центральных частот – 5 МГц, самая нижняя из них – 2405 МГц). Соответственно, скорость в каналах – 20 Кбит/с (в диапазоне 868 МГц), 40 Кбит/с (915 МГц) и 250 Кбит/с (2450 МГц). Стандарт IEEE 802.15.4 предусматривает небольшую дальность действия (около 10 метров).

В радиоканале использован метод широкополосной передачи с расширением спектра прямой последовательностью

(DSSS). Модуляция и расширяющие последовательности для диапазонов 868/915 и 2450 МГц различны.

Сеть стандарта IEEE 802.15.4 содержит два типа устройств – т.н. *полнофункциональные (FFD)* и устройства с *уменьшенной функциональностью (RFD)*. Их основное различие в том, что *FFD* могут устанавливать соединения с любыми устройствами, *RFD* – только с *FFD*. В каждой пикосети (*PAN*) должно быть устройство – координатор *PAN*. Его функции может выполнять только *FFD*.

Сеть, состоящая из одного *FFD* и нескольких *RFD*, образует топологию типа «звезда». Если в сети несколько *FFD*, топология может быть более сложной – типа «каждый с каждым» или представлять собой объединение нескольких звездообразных кластеров. Но в любом случае одно из *FFD* выполняет функцию *координатора сети*. Каждому устройству сети присваивается 64-разрядный адрес. Стандарт предусматривает взаимодействие устройств не только в рамках одной *PAN*, но и между различными соседними *PAN*, поэтому и необходима развитая система адресации. Для упрощения обмена внутри сети координатор *PAN* может присваивать устройствам более короткие 16-разрядные адреса. В этом случае для межсетевого взаимодействия используются 16-разрядные идентификаторы сетей, также назначаемые координатором.

Информационный обмен в пикосети происходит посредством последовательности *суперфреймов*. В общем случае *суперфрейм* включает управляющий интервал (*beacon*), за ним следует интервал конкурентного доступа (*CAP*), в соответствии с механизмом CSMA/CA, и период назначенного доступа. Последний содержит набор временных интервалов, назначенных определенным устройствам, чувствительным к задержкам, для передачи данных (*гарантированные тайм-слоты, GTS*) – например, для связи беспроводного манипулятора «мышь» с компьютером. Управляющий интервал передает только координатор *PAN*. Однако в суперфрейме может не быть ни управляющего интервала, ни *GTS*. В общем, структура суперфреймов аналогична принятой в стандарте IEEE 802.15.3.

Каждое устройство передает информацию посредством *фреймов (пакетов)*. Они могут быть четырех типов – *управляющие (beacon frame)*, *фреймы данных*, *фреймы подтверждения приема данных* и *фреймы команд MAC-уровня*. Фреймы физического уровня (рис. 6.4) содержат заголовок с синхропоследовательностью и информацией о размере фрейма (до 127 байт) и собственно поле данных – пакет MAC-уровня.

Заголовок синхронизации		Заголовок физического уровня		Поле данных
Преамбула	Маркер начала фрейма	Длина фрейма	Зарезервировано	
4 байта	1 байт	7 бит	1 бит	Произвольно

Рис. 6.4. Фреймы физического уровня

Последний содержит заголовок со всей необходимой информацией о фрейме (тип, наличие криптозащиты, необходимость подтверждения приема и т.п.), адреса и идентификаторы устройства - отправителя и получателя, поле данных и проверочную контрольную сумму (рис.6.5). Сама процедура обмена информацией может использовать пакеты подтверждения приема данных (если потеря пакета критична).

Заголовок МАС-уровня						Поле данных	Проверочная последовательность
Контроль кадра	Номер последовательности	Идентификатор сети назначения	Идентификатор устройства назначения	Идентификатор сети источника	Адрес источника		
		Адресные поля					
2 байта	1 байт	0/2 байта	0/2/8 байт	0/2 байта	0/2/8 байт	Произвольно	2 байта

Рис. 6.5. Структура кадра MAC-уровня

Стандарт IEEE 802.15.4a (UWB). В стандарте *UWB (Ultrawideband, сверхширокополосная связь)* используется самый широкий из распространенных сегодня технологий диапазон частот. Эта беспроводная технология предназначена для передачи данных в широком диапазоне частот (3,1–10,6 ГГц) на короткие

(до 5–10 м) расстояния с высокой пропускной способностью (400–500 Мбит/с) и низкой потребляемой мощностью (0,05 мВт). С помощью *Ultra Wideband* технологии можно строить специальные сети. В такой сети несколько сверхширокополосных устройств могут организовать связь между любыми узлами. Одно из основных преимуществ этой технологии заключается в том, что она не создает помех для других (таких как Wi-Fi, WiMAX) беспроводных технологий.

6.3. Беспроводной широкополосный доступ Broadband Wireless Access

Сегодня набирает популярность новая концепция широкополосного доступа в Интернет. Технология беспроводного широкополосного доступа может решить проблему последней мили в больших городах.

К городским беспроводным сетям (WMAN) относятся сети стандартов IEEE 802.16 (WiMAX) и 802.20 (Mobile Fi). Поскольку данные технологии используются в качестве стандартов беспроводного широкополосного доступа BWA, то технологию WiMAX также обозначают как BWA 802.16, а стандарт 802.20 (*Mobile Fi*) – как BWA-m (802.20).

WiMAX (Worldwide Interoperability for Microwave Access, микроволновый доступ в мировом масштабе) – телекоммуникационная технология, разработанная с целью предоставления универсальной беспроводной связи на больших расстояниях, за основу взят стандарт IEEE 802.16.

Каждая из спецификаций WiMAX (*фиксированный* и *мобильный* вариант) определяет свои рабочие диапазоны частот, ширину полосы пропускания, мощность излучения, методы передачи и доступа, способы кодирования и модуляции сигнала, принципы повторного использования радиочастот и прочие показатели. Основное различие двух технологий состоит в том, что *фиксированный* WiMAX позволяет обслуживать только «статичных» абонентов, а *мобильный* ориентирован на работу с пользователями, передвигающихся со скоростью до 120 км/ч. Мобильность означает наличие функций роуминга и «бесшовного» переключения между базовыми станциями при

передвижении абонента. В частном случае *мобильный* WiMAX может применяться и для обслуживания фиксированных пользователей.

В общем виде WiMAX сети состоят из основных частей: базовых и абонентских станций, оборудования, связывающего базовые станции между собой, с поставщиком сервисов и с Интернетом.

Для соединения базовой станции с абонентской используется высокочастотный диапазон радиоволн от 1,5 до 11 ГГц. В идеальных условиях скорость обмена данными может достигать 70 Мбит/с, при этом не требуется обеспечения прямой видимости между базовой станцией и приемником.

Между базовыми станциями устанавливаются соединения, использующие диапазон частот от 10 до 66 ГГц, скорость обмена данными может достигать 120 Мбит/с. При этом, по крайней мере, одна базовая станция подключается к сети провайдера с использованием классических проводных соединений.

Кадр стандарта 802.16 (рис.6.6) состоит из заголовка (48 бит или 6 байт), поля «Данные» и поля «Контрольная сумма кадра» (4 байта).



Рис. 6.6. Формат кадра стандарта IEEE 802.16

Бит «HT» показывает тип заголовка. При обычном MAC-кадре «HT=0». Бит «EC» указывает, шифруется ли поле данных. Если «EC=1», то поле данных зашифровано, а если «EC=0», то не зашифровано. Поле «T» (Type) состоит из 6 бит и указывает тип кадра. Бит «CI» указывает на наличие/отсутствие поля контрольной суммы кадра. Если «CI=1», то поле контрольной суммы кадра добавлено к полю данных, а если «CI=0», то – нет. Поле «EKS» (2 бита) сообщает, какой из ключей шифрования используется (только при «EC=1»). Поле

«*L*» (*Length*) состоит из 11 бит. В поле содержится информация о полной длине кадра, включая заголовок. Поле «*CID*» (*Connection ID, идентификатор соединения*), состоящее из 16 бит, показывает, какому из соединений принадлежит кадр. Поле «*Контрольная сумма заголовка*» предназначено для контроля целостности заголовка кадра и его значение вычисляется с помощью полинома $x^8 + x^2 + x + 1$.

Стандарт беспроводной связи IEEE 802.16 изначально был задуман для организации работы беспроводных сетей, покрывающих большие городские территории, поэтому его часто называют стандартом для *WMAN* (*Metropolitan Area Network – сеть для городских регионов*).

Стандарт IEEE 802.16 (2001 г.) описывает протоколы для городских сетей (WMAN) систем точка-многоточка. В стандарте определяется два рабочих диапазона частот - 10-66 ГГц и ниже 10 ГГц и содержится описание как уровня *MAC*, так и физического *PHY* уровня. На более высоких частотах приемное и передающее оборудование, работающие по этому стандарту, может находиться только в зоне прямой видимости. В диапазоне ниже 10 ГГц требование наличия прямой видимости не обязательно. Пропускная способность до 135 Мбит/с. Метод модуляции OFDM – 64-QAM.

Стандарт 802.16a. Стандарт 802.16a (2003 г.) является расширение базового стандарта и называется Wi-MAX. Известен так же как 802.16-2004 и *фиксированный WiMAX*. В стандарте 802.16a предусмотрен более низкий частотный диапазон – от 2 до 11 ГГц. Зона покрытия составляет 50 км, а максимальная скорость передачи данных – до 70 Мбит/с на один сектор базовой станции. Типовая базовая станция имеет до 6 секторов. Возможна работа в условиях отсутствия прямой видимости на базовую станцию. В большинстве стран под эту технологию отведены диапазоны 3,5 и 5 ГГц.

Стандарт 802.16e. Стандарт 802.16e предназначен для мобильных систем. Безопасность в сети обеспечивается на уровне протокола 3DES. Известен так же как 802.16-2005 и *мобильный WiMAX*. Стандарт принят в 2005 г. и базируется на технологии OFDMA. Возможна работа при наличии/отсутствии

прямой видимости. Частотные диапазоны для сетей Mobile WiMAX: 2,3–2,5; 2,5–2,7; 3,4–3,8 ГГц.

Стандарт IEEE 802.20. Стандарт для мобильного доступа к данным (IEEE 802.20) отличается высокой скоростью передачи данных (более 1 Мбит/с) и позволяет получать мобильный доступ к данным из движущихся транспортных средств, со скоростью не превышающей 250 км/ч.

WMAN обеспечивает доступ к сети фиксированной связи посредством установленных на зданиях антенн, взаимодействующих с центральными базовыми радиостанциями. В пределах зданий ставка делается на имеющиеся стандарты: IEEE 802.3 и IEEE 802.11. По причине высокой мобильности и эстафетной передачи абонента от одной базовой станции к другой эту технологию называют *беспроводной глобальной сетью WWAN (Wireless Wide Area Network)*.

Выводы:

В технологии FDDI в качестве физической среды впервые был использован волоконно-оптический кабель и достигнута скорость 100 Мбит/с. Высокая степень отказоустойчивости обеспечивается за счет применения двойного оптоволоконного кольца.

Стандарты IEEE 802.11 являются основными стандартами беспроводных локальных сетей. Существует несколько вариантов спецификаций физического уровня 802.11, отличающихся диапазоном используемых частот (2,4 и 5 ГГц), а также методом кодирования (FHSS, DSSS, OFDM).

Метод доступа 802.11 является комбинацией случайного метода доступа с предотвращением коллизий (DCF) и централизованного детерминированного метода доступа с опросом (PCF). Гибкое применение режимов DCF и PCF позволяет обеспечить поддержку показателей QoS для синхронного и асинхронного трафиков.

Персональные сети (PAN) предназначены для взаимодействия принадлежащих одному владельцу устройств на небольшом расстоянии, в радиусе от 10 до 100 м. Персональные сети должны обеспечивать как фиксированный доступ, так и

мобильный, когда владелец устройств перемещается вместе с ними между помещениями или городами.

Самой популярной технологией PAN является Bluetooth, в которой используется концепция пикосети, объединяющей до 255 устройств, но только 8 из них могут в каждый момент времени быть активными.

Для чувствительного к задержкам трафика сеть Bluetooth поддерживает синхронные каналы, ориентированные на соединение (SCO), а для эластичного – асинхронные каналы, не ориентированные на соединение (ACL).

Контрольные вопросы:

1. Чем отличается беспроводная сеть?
2. Какие типы данных передаются с помощью беспроводной сети?
3. Какие разновидности беспроводных сетей существуют?
4. Какие стандарты регламентируют принцип работы беспроводной сети?
5. Какова зона действия беспроводной персональной сети?
6. В каком частотном диапазоне работают Bluetooth-устройства?

7. ПРИНЦИПЫ ПОСТРОЕНИЯ ИНТЕРСЕТИ (INTERNETWORK)

7.1. Архитектура интегрированной сети

Информационная сеть в общем случае рассматривается как совокупность нескольких сетей и называется *составной сетью*, или *интерсетью* (*internetwork*, или *internet*). Сети, входящие в составную сеть, называются *подсетями* (*subnet*), *составляющими сетями* или просто *сетями* (рис. 7.1).

Подсети соединяются между собой маршрутизаторами. Компонентами составной сети могут быть как локальные, так и глобальные сети. Все узлы в пределах одной подсети взаимодействуют, используя единую для них технологию

(локальные Ethernet, Token Ring, FDDI и глобальные сети X.25, Frame Relay, ISDN). Каждая из этих технологий способна обеспечить взаимодействие всех узлов в своей подсети, но не способна построить информационную связь между произвольно выбранными узлами, принадлежавших разным подсетям, например между узлом «А» и узлом «В» на рис. 7.1. Следовательно, для организации взаимодействия между любой произвольной парой узлов «большой» составной сети требуются дополнительные средства, предоставляемые сетевым уровнем.

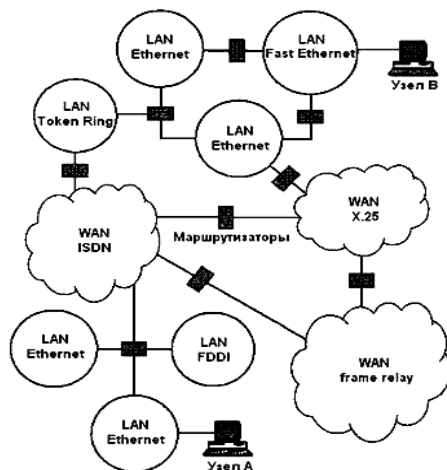


Рис. 7.1. Архитектура составной сети

Сетевой уровень выступает координатором, организующим работу всех подсетей, лежащих на пути продвижения пакета по составной сети. Для перемещения данных в пределах подсетей сетевой уровень обращается к используемым в этих подсетях технологиям.

Многие технологии локальных сетей (Ethernet, Token Ring, FDDI, Fast Ethernet и др.) используют одну и ту же систему адресации узлов на основе MAC-адресов, но существуют технологии (X.25, ATM, Frame Relay), в которых применяются другие схемы адресации. Адреса, присвоенные узлам в соответствии с локальными технологиями подсетей, называют *локальными*. Для выполнения своей задачи сетевой уровень должен иметь собственную глобальную систему адресации, не зависящую от способов адресации узлов в отдельных подсетях, которая позволила бы на сетевом уровне универсальным и однозначным способом идентифицировать любой узел составной сети.

Естественным способом формирования сетевого адреса является уникальная нумерация всех подсетей составной сети и нумерация всех узлов в пределах каждой подсети. Таким образом, сетевой адрес представляет собой пару: номер сети (подсети) и номер узла.

В качестве номера узла может выступать: локальный адрес этого узла (такая схема принята в стеке IPX/SPX); некоторое число, никак не связанное с локальной технологией, однозначно идентифицирующее узел в пределах данной подсети. В первом случае сетевой адрес становится зависимым от локальных технологий, и ограничивает его применение. Например, сетевые адреса IPX/SPX рассчитаны на работу в составных сетях, объединяющих сети, использующих только MAC-адреса или адреса аналогичного формата. Второй подход более универсален, он характерен для стека TCP/IP. В том и другом случае каждый узел составной сети имеет наряду со своим локальным адресом еще один – универсальный сетевой адрес.

Данные, поступающие на сетевой уровень и которые необходимо передать через составную сеть, снабжаются заголовком сетевого уровня. Совместно с заголовком они образуют пакет. Заголовок пакета сетевого уровня имеет

унифицированный формат, не зависящий от форматов кадров канального уровня тех сетей, входящих в составную сеть, и несет, наряду с другой служебной информацией, данные о номере сети назначения этого пакета. Сетевой уровень определяет маршрут и перемещает пакет между подсетями.

Каждый раз, при передаче пакет сетевого уровня из одной подсети в другую, он извлекается из кадра первой подсети (освобождается от канального заголовка этой сети) и упаковывается в кадр (снабжается новым заголовком) канального уровня следующей подсети. Информацию, на основе которой производится эта замена, предоставляют служебные поля пакета сетевого уровня. В поле адреса назначения нового кадра указывается локальный адрес следующего маршрутизатора.

Если провести аналогию между взаимодействием разнородных сетей и перепиской людей из разных стран, то сетевая информация – это общепринятый индекс страны, добавляемый к адресу письма, написанному на одном из сотни языков земного шара, например на санскрите. И даже если это письмо должно пройти через множество стран, почтовые работники не знающие санскрит, понятный им индекс страны-адресата подскажет, через какие промежуточные пункты лучше передать письмо, чтобы оно кратчайшим путем попало в Индию. А работники местных почтовых отделений смогут прочесть точный адрес – адрес идентифицирующий город, улицу, дом и адресата, и доставить письмо, т. к. адрес написан на языке и в форме, принятой в данной стране.

Основным полем заголовка сетевого уровня является номер сети – адресата. В ранних протоколах локальных сетей такого поля в кадрах не было – предполагалось, что все узлы принадлежат одной сети. Явная нумерация сетей позволяет протоколам сетевого уровня составлять точную карту межсетевых связей и выбирать рациональные маршруты при любой их топологии, в том числе альтернативные маршруты, при их наличии.

Кроме номера сети заголовок сетевого уровня должен содержать и информацию, необходимую для успешного перехода пакета из сети одного типа в сеть другого типа:

- номер фрагмента пакета, необходимый для успешного проведения операций сборки/разборки фрагментов при соединении сетей с разными максимальными размерами пакетов;
- время жизни пакета, указывающее, на время продвижения по интернету, оно может использоваться для уничтожения «заблудившихся» пакетов;
- качество услуги - критерий выбора маршрута при межсетевых передачах: например, узел-отправитель может потребовать передать пакет с максимальной надежностью, возможно, в ущерб времени доставки.

Когда две или более сети организуют совместную транспортную службу, то такой режим взаимодействия обычно называют *межсетевым взаимодействием (internet working)*.

7.2. Принципы маршрутизации

Важнейшей задачей сетевого уровня является маршрутизация – передача пакетов между двумя конечными узлами в составной сети.

Рассмотрим принципы маршрутизации на примере составной сети (рис. 7.2). В этой сети 20 маршрутизаторов объединяют 18 сетей в общую сеть: S1, S2,..., S20 – это номера сетей. Маршрутизаторы имеют по несколько портов (минимум по два), к которым присоединяются сети. Каждый порт маршрутизатора можно рассматривать как отдельный узел сети: он имеет собственный сетевой адрес и собственный локальный адрес в той подсети, которая к нему подключена. Например, маршрутизатор 1 имеет три порта, к которым подключены сети S1, S2, S3. На рис. 7.2. сетевые адреса этих портов обозначены как M1 (1), M1 (2) и M1 (3). Порт M1 (1) имеет локальный адрес в сети с номером S1, порт M1 (2) – в сети S2, а порт M1 (3) – в сети S3. Таким образом, маршрутизатор можно рассматривать как совокупность нескольких узлов, каждый из которых входит в свою сеть. Как единое устройство маршрутизатор не имеет ни отдельного сетевого адреса, ни какого-либо локального адреса.

В сложных составных сетях почти всегда существует несколько альтернативных маршрутов для передачи пакетов между двумя конечными узлами. *Маршрут* - это

последовательность маршрутизаторов, через которые должен пройти пакет от отправителя до получателя. Так, пакет, отправленный из узла «А» в узел «В», может пройти через маршрутизаторы 17, 12, 5, 4 и 1 или маршрутизаторы 17, 13, 7, 6 и 3. Нетрудно найти еще несколько маршрутов между узлами «А» и «В».

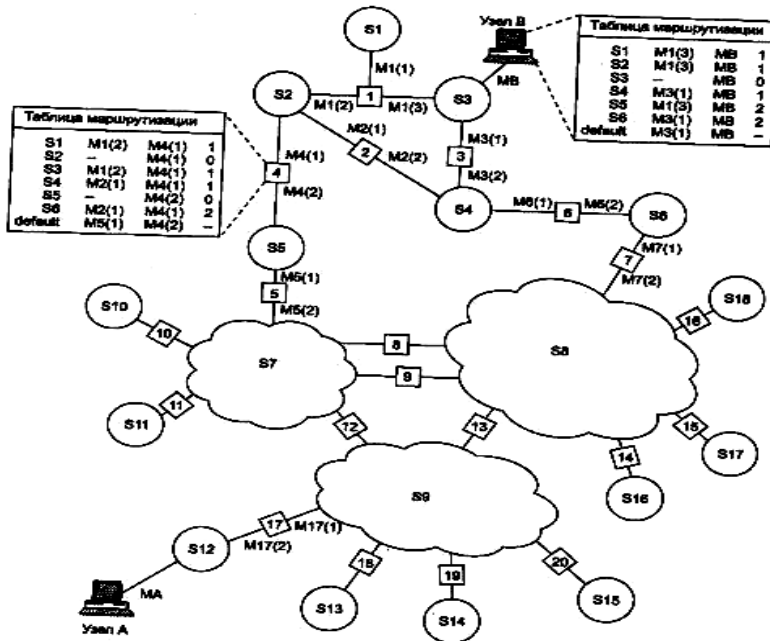


Рис. 7.2. Принципы маршрутизации в составной сети

Задачу выбора маршрута из нескольких возможных решают не только маршрутизаторы, а также и конечные узлы. Маршрут выбирается на основании имеющейся у этих устройств информации о текущей конфигурации сети, а также на основании указанного критерия выбора маршрута. Обычно в качестве критерия выступают: задержка прохождения маршрута отдельным пакетом или средняя пропускная способность маршрута для последовательности пакетов.

7.3. Протоколы маршрутизации

Задача маршрутизации решается на основе анализа таблиц маршрутизации, размещенных в маршрутизаторах и конечных узлах сети. Основная работа по созданию таблиц маршрутизации выполняется автоматически, но имеется и возможность вручную скорректировать или дополнить таблицу.

Для автоматического построения таблиц маршрутизаторы обмениваются информацией о топологии составной сети в соответствии со специальным служебным протоколом. Протоколы этого типа называются протоколами маршрутизации (или маршрутизирующими протоколами). Протоколы маршрутизации (например, RIP, OSPF, NLSP) отличаются от сетевых протоколов (например, IP, IPX). И те и другие выполняют функции сетевого уровня модели *OSI*. Но в то время как первые собирают и передают по сети чисто служебную информацию, вторые предназначены для передачи пользовательских данных. Протоколы маршрутизации используют сетевые протоколы как транспортное средство. При обмене маршрутной информацией пакеты протокола маршрутизации помещаются в поле данных пакетов сетевого уровня или даже транспортного уровня.

Маршрутизаторы для принятия решения о продвижении пакета обращаются к адресным таблицам, но вместо MAC-адресов в таблицах маршрутизации указываются номера сетей, которые соединяются в интернет. Другим отличием таблиц маршрутизации от адресных таблиц мостов является способ их создания. В то время как мост строит таблицу, пассивно наблюдая за проходящими через него информационными кадрами, посылаемыми конечными узлами сети друг другу, маршрутизаторы по своей инициативе обмениваются специальными служебными пакетами, сообщая соседям об известных им сетях в интернет, маршрутизаторах и о связях этих сетей с маршрутизаторами. Обычно учитывается не только топология связей, но и их пропускная способность и состояние.

При изменении конфигурации сети некоторые записи в таблице становятся недействительными. От того, насколько быстро протокол маршрутизации приводит в соответствие

содержимое таблицы реальному состоянию сети, зависит качество работы всей сети.

При выборе рационального маршрута определялся только ближайший маршрутизатор, а не вся последовательность маршрутизаторов от начального до конечного узла. В соответствии с этим подходом маршрутизация выполняется по распределенной схеме – каждый маршрутизатор ответственен за выбор только одного шага маршрута. Такие алгоритмы маршрутизации называются *одношаговыми*.

Существует и прямо противоположный, *многошаговый подход* – *маршрутизация от источника (Source Routing)*. В соответствии с ним узел-источник задает в отправляемом в сеть пакете полный маршрут его следования через все промежуточные маршрутизаторы. При использовании такой маршрутизации нет необходимости строить и анализировать таблицы маршрутизации. Это ускоряет прохождение пакета по сети, разгружает маршрутизаторы, но при этом большая нагрузка ложится на конечные узлы. Эта схема в локальных сетях применяется сегодня гораздо реже, чем схема распределенной одношаговой маршрутизации. Однако в новой версии протокола IP наряду с классической одношаговой маршрутизацией разрешена и маршрутизация от источника.

Одношаговые алгоритмы в зависимости от способа формирования таблиц маршрутизации делятся на три класса:

- алгоритмы фиксированной (или статической) маршрутизации;
- алгоритмы простой маршрутизации;
- алгоритмы адаптивной (или динамической) маршрутизации.

В алгоритмах *фиксированной маршрутизации* все записи в таблице маршрутизации являются статическими. Администратор сети сам решает, на какие маршрутизаторы надо передавать пакеты с теми или иными адресами, и вручную заносит соответствующие записи в таблицу маршрутизации. Таблица, как правило, создается в процессе загрузки, в дальнейшем она используется без изменений до тех пор, пока ее содержимое не будет отредактировано вручную. Различают одномаршрутные таблицы, в которых для каждого адресата задан один путь, и многомаршрутные таблицы, определяющие несколько альтернативных путей для каждого адресата. В

многомаршрутных таблицах должно быть задано правило выбора одного из маршрутов. Чаще всего один путь является основным, а остальные – резервными. Алгоритм фиксированной маршрутизации приемлем только в небольших сетях с простой топологией. Однако этот алгоритм может эффективно использоваться и для работы на магистралях крупных сетей, так как сама магистраль может иметь простую структуру.

В алгоритмах *простой маршрутизации* таблица маршрутизации может не использоваться, или строится без помощи протоколов маршрутизации. Выделяют три типа простой маршрутизации:

- *случайная маршрутизация*, когда прибывший пакет посылается в первом попавшем случайном направлении, кроме исходного;

- *лавинная маршрутизация*, когда пакет широковещательно посылается по всем возможным направлениям, кроме исходного;

- *маршрутизация по предыдущему опыту*, когда выбор маршрута осуществляется по таблице, но таблица строится по принципу моста путем анализа адресных полей пакетов, появляющихся на входных портах.

Самыми распространенными являются алгоритмы *адаптивной маршрутизации*. Эти алгоритмы обеспечивают автоматическое обновление таблиц маршрутизации после изменения конфигурации сети. Протоколы, построенные на их основе, позволяют всем маршрутизаторам собирать информацию о топологии связей в сети, оперативно отрабатывая все изменения конфигурации связей. В таблицах маршрутизации при этой маршрутизации обычно имеется информация об интервале времени, в течение которого данный маршрут будет оставаться действительным. Это время называют *временем жизни маршрута (Time To Live, TTL)*.

Адаптивные алгоритмы имеют распределенный характер, выражаемый в отсутствии в сети каких-либо выделенных маршрутизаторов, собирающих и обобщающих топологическую информацию: эта работа распределена между всеми маршрутизаторами.

Адаптивные алгоритмы маршрутизации должны отвечать нескольким важным требованиям:

- обеспечивать рациональность маршрута;
- быть достаточно простыми: не требовать большого объема вычислений и порождать интенсивный служебный трафик;
- обладать свойством сходимости, т. е. всегда приводить к однозначному результату за приемлемое время.

Адаптивные протоколы обмена маршрутной информацией делятся на две группы:

- дистанционно-векторные алгоритмы (*Distance Vector Algorithms, DVA*);
- алгоритмы состояния связей (*Link State Algorithms, LSA*).

В алгоритмах *дистанционно-векторного типа* каждый маршрутизатор периодически и широковещательно рассылает по сети вектор, компонентами которого являются расстояния от данного маршрутизатора до всех известных ему сетей (число хопов). При получении вектора от соседа маршрутизатор наращивает расстояния до указанных в векторе сетей на расстояние до данного соседа. Получив вектор от соседнего маршрутизатора, каждый маршрутизатор добавляет к нему информацию об известных ему других сетях, о которых он узнал непосредственно или из аналогичных объявлений других маршрутизаторов, а затем снова рассылает новое значение вектора по сети. В итоге, каждый маршрутизатор узнает информацию обо всех имеющихся в интерсети сетях и о расстоянии до них через соседние маршрутизаторы.

Дистанционно-векторные алгоритмы хорошо работают только в небольших сетях. Так как они засоряют линии связи интенсивным широковещательным трафиком, к тому же изменения конфигурации могут отрабатываться по этому алгоритму не всегда корректно, т. к. маршрутизаторы не имеют точного представления о топологии связей в сети. Его работа в соответствии с дистанционно-векторным протоколом напоминает работу моста, так как точной топологической картины сети такой маршрутизатор не имеет.

Наиболее распространенным протоколом, основанным на дистанционно-векторном алгоритме, является протокол RIP,

который распространен в двух версиях – RIP IP, работающий с протоколом IP, и RIP IPX, работающий с протоколом IPX.

Алгоритмы состояния связей обеспечивают каждый маршрутизатор информацией, достаточной для построения точного графа связей сети. «Широковещательная» рассылка используется здесь только при изменениях состояния связей, что происходит в надежных сетях не так часто. Вершинами графа являются как маршрутизаторы, так и объединяемые ими сети.

Чтобы понять, в каком состоянии находятся линии связи, подключенные к его портам, маршрутизатор периодически обменивается короткими пакетами «HELLO» со своими ближайшими соседями. Этот служебный трафик также засоряет сеть, но не в такой степени как RIP-пакеты, т. к. пакеты «HELLO» имеют намного меньший объем.

Протоколами, основанными на *алгоритме состояния связей*, являются протоколы *IS-IS (Intermediate System to Intermediate System)* стека *OSI*, *OSPF (Open Shortest Path First)* стека *TCP/IP* и *протокол NLSP стека Novell*.

Выводы:

Информационная сеть в общем случае рассматривается как составная сеть, или интернет (internetwork, или internet), содержащая подсети (subnet).

Подсети соединяются между собой маршрутизаторами. Протоколы маршрутизации генерируют для каждого маршрутизатора согласованные таблицы маршрутизации, позволяющие обеспечивать доставку пакета по рациональному маршруту от исходной сети в сеть назначения за конечное число шагов.

При маршрутизации выполняемой по распределенной схеме: каждый маршрутизатор ответственен за выбор только одного шага маршрута, алгоритмы маршрутизации называются одношаговыми.

При маршрутизации выполняемой по заданному полному маршруту следования алгоритмы маршрутизации называются с многошаговым подходом –маршрутизация от источника (Source Routing).

Одношаговые алгоритмы самые востребованные и в зависимости от способа формирования таблиц маршрутизации делятся на три класса:

- алгоритмы статической маршрутизации;
- алгоритмы простой маршрутизации;
- алгоритмы динамической маршрутизации.

Динамическая маршрутизация обеспечивает автоматическое обновление таблиц маршрутизации после изменения конфигурации сети.

Динамические протоколы маршрутизации делятся на дистанционно-векторные алгоритмы (например, RIP) и алгоритмы состояния связей (например, OSPF).

Контрольные вопросы:

1. Какова архитектура интегрированной сети?
2. Что такое маршрутизация и каковы ее виды?
3. В чем заключаются принципы маршрутизации?
4. Какие протоколы маршрутизации используются?
5. Дать характеристику устройству «маршрутизатор».
6. Каковы функции маршрутизатора?
7. Какие существуют алгоритмы маршрутизации?
8. Чем отличаются фиксированная, простая маршрутизация и адаптивная маршрутизации?
9. Лавинная и случайная маршрутизации?
10. Какую информацию должен содержать заголовок сетевого уровня необходимую для успешного перехода пакета из одного типа сети в сеть другого типа?
11. Для какой цели применяется параметр время жизни пакета?
12. Какой режим взаимодействия называют межсетевым взаимодействием?
13. При какой маршрутизации используется время жизни маршрута?
14. Каким важным требованиям должны отвечать адаптивные алгоритмы маршрутизации?
15. Какие протоколы работают по дистанционно-векторным алгоритмам?
16. Какие протоколы работают по алгоритмам состояния связей?

8. АДРЕСАЦИЯ В СТЕКЕ ПРОТОКОЛОВ TCP/IP

8.1. Стек протоколов TCP/IP

В модели *TCP/IP* (*Transmission Control Protocol*, протокол управления передачей и *Internet protocol*, межсетевой протокол) прикладной уровень соответствует прикладному, уровню представления и сеансовому, а уровень доступа к сети (*Host-to-network*) – физическому и каналному уровням *OSI* (рис. 8.1).

Модель OSI		Модель TCP/IP	
7 Прикладной	Application	4	Прикладной Application
6 Представления	Presentation		
5 Сеансовый	Session		
4 Транспортный	Transport	3	Транспортный Transport
3 Сетевой	Network	2	Уровень межсетевого взаимодействия Internet (межсетевой)
2 Канальный	Data Link		
1 Физический	Physical	1	Уровень доступа к сети (сетевой) Host-to-network

Рис. 8.1. Структура уровней OSI и TCP/IP

В стеке *TCP/IP* определены 4 уровня: *прикладной*, *транспортный*, *уровень межсетевого взаимодействия*, *уровень сетевых интерфейсов*. Каждый из этих уровней несет на себе некоторую нагрузку по решению основной задачи - организации надежной и производительной работы составной сети, части которой построены на основе разных сетевых технологий.

Стек *TCP/IP*, в отличие от семиуровневой модели *OSI*, принято описывать в рамках четырех уровней, на которых функционируют протоколы (рис. 8.2).

Прикладной (application) уровень стека TCP/IP объединяет службы, предоставляемые системой пользовательским приложениям. За долгие годы использования в сетях различных стран и организаций стек TCP/IP накопил большое количество протоколов и служб прикладного уровня. К ним относятся такие широко используемые протоколы: протокол копирования файлов (*File Transfer Protocol*, *FTP*); протокол эмуляции терминала

(telnet); простой протокол передачи электронной почты (*Simple Mail Transfer Protocol, SMTP*); протокол передачи гипертекстовой информации (*Hypertext Transfer Protocol, HTTP*) и мн. др.

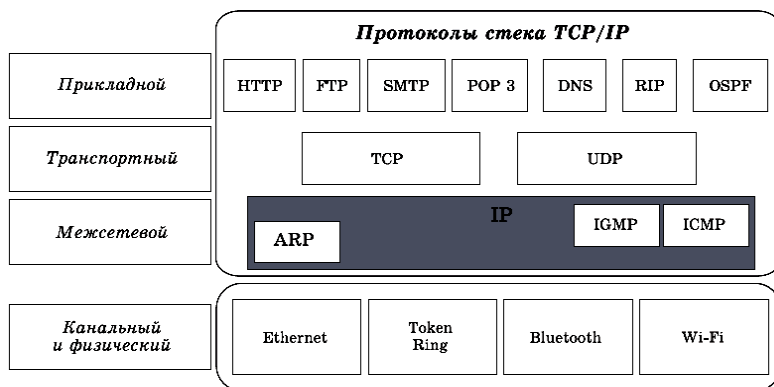


Рис. 8.2. Основные протоколы стека TCP/IP

Протоколы прикладного уровня устанавливаются на хостах, уровень реализуется программными системами, построенными по архитектуре клиент-сервер. В отличие от протоколов остальных уровней, протоколы данного уровня отрабатывают логику приложений и «не интересуются» способами передачи данных по сети, обращаются к протоколам нижних уровней как к некоторому набору инструментов. Так, клиентская часть протокола для обмена сообщениями со своей серверной частью, установленной на отдаленном узле составной сети, должна обратиться с запросом к нижележащему транспортному уровню.

Транспортный (transport) уровень стека TCP/IP, называемый *основным* уровнем, может предоставлять вышележащему уровню два типа сервиса:

- гарантированная доставка – протокол управления передачей (*Transmission Control Protocol, TCP*);
- доставка «по возможности» (*«best effort»*) – протокол пользовательских дейтаграмм (*User Datagram Protocol, UDP*).

Для обеспечения надежной доставки данных, протокол ТСП предусматривает установление логического соединения, это позволяет ему нумеровать пакеты, подтверждать их прием квитанциями, в случае потери организовывать повторные передачи, распознавать и уничтожать дубликаты, доставлять прикладному уровню пакеты в том порядке, в котором они были отправлены. Этот протокол позволяет равноранговым объектам на компьютере-отправителе и компьютере-получателе поддерживать обмен данными в дуплексном режиме.

ТСП позволяет без ошибок доставить сформированный на одном из компьютеров поток байтов в любой другой компьютер, входящий в составную сеть, делит поток байтов на части – *сегменты* и передает их нижележащему уровню межсетевого взаимодействия. После того как эти сегменты будут доставлены средствами уровня межсетевого взаимодействия в пункт назначения, протокол ТСП снова соберет их в непрерывный поток байтов.

Второй протокол этого уровня – UDP является простейшим дейтаграммным протоколом, который используется в том случае, когда задача надежного обмена данными либо вообще не ставится, либо решается средствами более высокого уровня – системными прикладными службами или пользовательскими приложениями.

В функции протоколов транспортного уровня ТСП и UDP входит также выполнение роли связующего звена между прилегающим к ним прикладным уровнем и уровнем межсетевого взаимодействия. От прикладного протокола транспортный уровень принимает задание на передачу данных с тем или иным качеством, а после выполнения рапортует об этом прикладному уровню. К нижележащему уровню межсетевого взаимодействия протоколы ТСП и UDP обращаются как к инструменту, не очень надежному, но способному перемещать в свободном и рискованном путешествии по составной сети. Протоколы ТСП и UDP, так же как и протоколы прикладного уровня, устанавливаются на хостах.

Уровень межсетевого взаимодействия (internet), называемый **сетевым (network) уровнем**, является стержнем всей архитектуры *TCP/IP*. Именно уровень, функции которого

соответствуют сетевому уровню модели *OSI*, обеспечивает перемещение пакетов в пределах всей составной сети. Протоколы уровня межсетевого взаимодействия поддерживают интерфейсы с вышележащим транспортным уровнем, получая от него запросы на передачу данных по сети.

8.2. Типы адресов стека TCP/IP

Принятый в IP-сетях способ адресации узлов, в немалой степени, способствует масштабируемости данной технологии, которая позволяет однозначно идентифицировать миллионы сетевых интерфейсов (вспомним хотя бы Интернет с его многомиллионной армией пользователей). Для обеспечения этой возможности в технологию *TCP/IP* включен целый ряд специальных механизмов и протоколов.

В стеке TCP/IP используются три типа адресов:

- *локальные* или *аппаратные* адреса, используемые для адресации узлов в пределах подсети;
- *сетевые* или *IP-адреса*, используемые для однозначной идентификации узлов в пределах всей составной сети;
- *доменные имена* – символьные идентификаторы узлов, к которым часто обращаются пользователи.

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

IP-адрес, состоящий из 4 байт, например, *109.26.17.100*. Этот адрес используется на сетевом уровне. Он назначается администратором во время конфигурирования компьютеров и

маршрутизаторов. IP-адрес состоит из двух частей: номера сети и номера узла. Номер сети может быть выбран администратором произвольно, либо назначен по рекомендации специального подразделения Internet (*Network Information Center, NIC*), если сеть должна работать как составная часть Internet. Провайдеры услуг Internet получают диапазоны адресов у подразделений *NIC*, а затем распределяют их между своими абонентами.

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

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

В общем случае сетевой интерфейс может иметь одновременно один или несколько локальных адресов и один или несколько сетевых адресов, а также одно или несколько доменных имен.

Итак, аппаратный (локальный) адрес идентифицирует узел в пределах подсети. Если подсеть использует одну из базовых технологий LAN – Ethernet, FDDI, Token Ring, то для доставки данных любому узлу такой подсети достаточно указать *MAC-адрес*. Таким образом, в этом случае аппаратным адресом является *MAC-адрес*.

В составную сеть *TCP/IP* могут входить подсети, построенные на основе более сложных технологий, к примеру, технологии IPX/SPX. Эта сеть сама может быть разделена на подсети и так же как IP-сеть, она идентифицирует свои узлы аппаратными и сетевыми IPX-адресами. Но поскольку для составной сети *TCP/IP* составная сеть IPX/SPX является обычной подсетью, в качестве аппаратных адресов узлов этой

подсети выступают те адреса, которые однозначно определяют узлы в данной подсети, а такими адресами являются IPX-адреса. Аналогично, если в составную сеть включена сеть X.25, то локальными адресами для протокола IP соответственно, будут адреса X.25.

Символьные имена в IP-сетях называются *доменными* и строятся по иерархическому признаку. Составляющие полного символьного имени в IP-сетях разделяются точкой и перечисляются в определенном порядке. Первое – простое имя хоста, второе – имя группы узлов (например, имя организации), затем – имя более крупной группы (поддомена) и так до имени домена самого высокого уровня (например, домена, объединяющего организации по географическому принципу: *RU* - Россия, *UK* - Великобритания, *US* - США). Примером доменного имени может служить имя *base2.sales.zil.ru*. Между доменным именем и IP-адресом узла нет никакой функциональной зависимости, поэтому единственный способ установления соответствия - это таблица. В сетях *TCP/IP* используется специальная распределенная служба доменных имен (*Domain Name System, DNS*), которая устанавливает это соответствие на основании создаваемых администраторами сети таблиц соответствия. Поэтому доменные имена называют также *DNS-именами*.

8.3. Формат записи IP-адреса

IP-адрес имеет длину 4 байта (32 бита) и состоит из двух логических частей - номера сети и номера узла в сети.

Наиболее употребляемой формой представления IP-адреса является запись в виде четырех чисел, представляющих значения каждого байта в десятичной форме и разделенных точками, например: **128.10.2.30**.

Этот же адрес может быть представлен в двоичном формате:

10000000000010100000001000011110.

А также в шестнадцатеричном формате: **80.0A.02.1D.**

Запись адреса не предусматривает специального разграничительного знака между номером сети и номером узла.

Для выделения маршрутизатором из 32 бит IP-адреса назначения пакетов номера сети, номера узла или определения дальнейший маршрута, используются несколько вариантов. Первый состоит в предварительном делении на две части не равной, но фиксированной длины 32- фиксированной длины, все сети будут иметь одинаковое максимальное число узлов. Если, например, под номер сети отвести один первый байт, то все адресное пространство распадется на сравнительно небольшое (2^8) число сетей огромного размера (2^{24} узлов). Если границу передвинуть дальше вправо, то сетей станет больше, но все равно все они будут одинакового размера. Очевидно, что такой жесткий подход не позволяет дифференцированно подходить к потребностям отдельных предприятий и организаций. Именно поэтому такой способ структуризации адреса и не нашел применения.

Второй подход основан на использовании маски, позволяющей максимально гибко устанавливать границу между номером сети и номером узла. В данном случае *маска* – это число, используемое в паре с IP-адресом; двоичная запись маски содержит последовательность единиц в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети. Поскольку номер сети является цельной частью адреса, единицы в маске также должны представлять непрерывную последовательность. Граница между последовательностью единиц и последовательностью нулей в маске соответствует границе между номером сети и номером узла в IP-адресе. При таком подходе адресное пространство можно представить как совокупность множества сетей разного размера.

И, традиционный способ решения данной проблемы заключается в использовании классов. Этот способ представляет собой компромисс между двумя вышеописанными способами: размеры сетей хотя и не являются произвольными, как при использовании масок, но и не являются одинаковыми, как при установлении фиксированных границ. Вводится несколько классов сетей, и для каждого класса определены свои размеры.

8.3.1. Классы IP-адресов

Традиционная схема деления IP-адреса на номер сети и номер узла основана на понятии класса, который определяется значениями нескольких первых битов адреса. Именно потому, что первый байт адреса *185.23.44.206* попадает в диапазон *128-191*, можно сказать, этот адрес относится к классу «В», а значит, номером сети являются первые два байта IP-адреса, дополненные двумя нулевыми байтами – *185.23.0.0*, а номером узла – два младших байта, дополненные с начала двумя нулевыми байтами – *0.0.44.206*.

Принадлежность IP-адреса к классу определяется значениями первых битов адреса. На рис. 8.3 показана структура IP-адресов разных классов

	1 байт		3 байта	
Класс А	0	№ сети	№ узла	
	2 байта		2 байта	
Класс Б	1	0	№ сети	№ узла
	3 байта			1 байт
Класс С	1	0	0	№ сети
				№ узла

Рис. 8.3. Структура IP-адресов

Если адрес начинается с 0, то этот адрес относится к классу «А», в котором под номером сети отводится один байт, а остальные три байта интерпретируются как номер узла в сети. Сети, имеющие номера в диапазоне от *1 (00000001)* до *126 (1111110)*, называются сетями класса «А». (Номер 0 не используется, а номер *127* зарезервирован для специальных целей). Сетей класса «А» немного, зато количество узлов в них может достигать 2^{24} , т. е. *16 777 216* узлов.

Если первые два бита адреса равны *10*, то адрес относится к классу «В». В адресах класса «В» под номер сети и под номер узла отводится по два байта. Сети, имеющие номера в диапазоне от *128.0 (10000000000000000000)* до *191.255 (10111111 11111111)*, называются сетями класса «В». Таким образом, сетей класса

«В» больше, чем сетей класса «А», но размеры их меньше, максимальное количество узлов в них составляет 2^{16} (65 536).

Если адрес начинается с последовательности битов *110*, то это адрес класса «С». В этом случае под номер сети отводится 24 бита, а под номер узла – 8 бит. Сети класса «С» наиболее распространены, но число узлов в них ограничено значением 2^8 (256) узлов. В табл. 8.1 приведены диапазоны номеров сетей и максимальное число узлов, соответствующих каждому классу сетей.

Таблица 8.1.
Характеристики адресов разного класса

Класс	Первые биты	Наименьший номер сети	Наибольший номер сети	Максимальное число узлов в сети
A	<i>0</i>	1.0.0.0	126.0.0.0	2^{24}
B	<i>10</i>	128.0.0.0	191.255.0.0	2^{16}
C	<i>110</i>	192.0.1.0	223.255.255.0	2^8
D	1110	224.0.0.0	239.255.255.255	Multicast
E	11110	240.0.0.0	247.255.255.255	Зарезервирован

8.3.2. Особые IP-адреса

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

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

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

Если все двоичные разряды IP-адреса равны 1, то пакет рассылается всем узлам, находящимся в той сети, что и источник этого пакета.

Если единицы стоят в поле номера узла, то пакет рассылается всем узлам сети с заданным номером сети (широковещательная рассылка).

127 – запрещенный номер для начала адресации (127.0.0.3).

8.4. Порядок назначения IP-адресов

У каждой подсети в пределах составной сети должен быть собственный уникальный номер, следовательно, процедура распределения номеров должна быть централизованной. Аналогично, узлы должны быть однозначно пронумерованы в пределах каждой из подсетей, отсюда следует, что централизованный характер должна иметь и процедура распределения номеров узлов в пределах каждой подсети. Если сеть небольшая, то уникальность адресов может быть обеспечена вручную администратором.

В больших сетях, подобных Интернету, уникальность сетевых адресов гарантируется централизованной, иерархически организованной системой их распределения. Главным органом регистрации глобальных адресов в Интернете с 1998 г. является *ICANN (Internet Corporation for Assigned Names and Numbers)* – неправительственная некоммерческая организация, управляемая советом директоров. Эта организация координирует работу региональных отделов, деятельность которых охватывает большие географические площади: ARIN (Америка), RIPE (Европа), APNIC (Азия и Тихоокеанский регион). Региональные отделы выделяют блоки адресов сетей крупным поставщикам услуг, те, в свою очередь, присваивают их своим клиентам, среди которых могут быть и более мелкие поставщики услуг.

Понятно, что в любой автономной сети могут быть использованы произвольные IP- адреса, лишь бы они были синтаксически правильными и уникальными в пределах этой сети. Совпадение адресов, в не связанных между собой сетях, не вызовет никаких отрицательных последствий. Однако во всех сетях, входящих в единую составную сеть (например, Интернет), должны использоваться глобально уникальные IP-адреса, т. е. адреса, однозначно определяющие сетевые интерфейсы в пределах всей составной сети.

Уже сравнительно давно наблюдается дефицит IP-адресов. Очень трудно получить адрес класса «B» и практически невозможно стать обладателем адреса класса «A». При этом надо отметить, что дефицит обусловлен не только ростом сетей и тем, что имеющееся адресное пространство используется

нерационально. Очень часто владельцы сетей класса «С» расходуют лишь небольшую часть из имеющихся у них 254 адресов.

Если же некоторая IP-сеть создана для работы в «автономном режиме», без связи с Интернетом, тогда администратор этой сети может назначить ей произвольно выбранный номер. Но и в этой ситуации для того, чтобы избежать каких-либо коллизий, в стандартах Интернета определено несколько адресов, рекомендуемых для автономного использования: в классе «А» – это сеть *10.0.0.0*, в классе «В» – 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 смягчения проблемы дефицита адресов разработчики стека TCP/IP предлагают разные подходы. Принципиальным решением является переход на новую версию IPv6, в которой резко расширяется адресное пространство за счет использования 16-байтных адресов. Однако и текущая версия IPv4 поддерживает некоторые технологии, направленные на более экономное расходование IP-адресов, одной из таких технологий является технология *бесклассовой междоменной маршрутизации (Classless Inter-Domain Routing, CIDR)*. Технология *CIDR* основана на масках, она отказывается от традиционной концепции разделения адреса IP-протокола IP на классы, что позволяет выдавать в пользование столько адресов – сколько реально необходимо потребителю. Благодаря *CIDR* поставщик получает возможность «нарезать» блоки из выделенного ему адресного пространства в точном соответствии с требованиями клиента.

8.5. Отображение IP-адресов на локальные адреса

Одной из главных задач протокола IP-обеспечение совместной согласованной работы в сети, состоящей из подсетей, использующих разные сетевые технологии.

Взаимодействие технологии *TCP/IP* с частными технологиями подсетей происходит многократно при перемещении пакета IP по составной сети. На каждом маршрутизаторе протокол IP определяет, в какую следующую

подсеть и какому пограничному узлу в этой подсети надо направить пакет. Таким пограничным узлом является маршрутизатор, и протоколу IP известен его IP-адрес. Очевидно, что для того, чтобы частная технология подсети смогла доставить пакет в последующий маршрутизатор, необходимо:

- во-первых, упаковать пакет в кадр соответствующего для данной подсети формата (например, Ethernet);
- во-вторых, снабдить кадр адресом, формат которого был бы понятен локальной технологии подсети (преобразовать, например, IP-адрес в MAC-адрес).

Решением этих задач занимается уровень сетевых интерфейсов стека TCP/IP.

Для определения локального адреса по IP-адресу используется *протокол разрешения адресов (Address Resolution Protocol, ARP)*.

Протокол разрешения адресов реализуется различным образом в зависимости от того, какой протокол канального уровня работает в данной сети. Протокол локальной сети (*Ethernet, Token Ring, FDDI*) с возможностью широковещательного доступа одновременно ко всем узлам сети или же какой-либо из протоколов глобальной сети (*X.25, Frame Relay*), которые, как правило, не поддерживают широковещательный доступ.

8.5.1. Протокол разрешения адресов

В протоколе IP-адрес узла, т. е. адрес компьютера или порта маршрутизатора, назначается произвольно администратором сети и прямо не связан с его локальным адресом. Подход, используемый в IP, удобно использовать в крупных сетях и по причине его независимости от формата локального адреса, и по причине стабильности, т. к. в противном случае, при смене на компьютере сетевого адаптера это изменение должны бы были учитывать все адресаты всемирной сети.

Локальный адрес используется в протоколе IP только в пределах локальной сети при обмене данными между маршрутизатором и узлом этой сети. Маршрутизатор, получив

пакет для узла одной из сетей, непосредственно подключенных к его портам, должен для передачи пакета сформировать кадр в соответствии с требованиями принятой в этой сети технологии и указать в нем локальный адрес узла, например его MAC-адрес. В пришедшем пакете этот адрес не указан, поэтому перед маршрутизатором встает задача поиска его по известному IP-адресу, который указан в пакете в качестве адреса назначения. С аналогичной задачей сталкивается и конечный узел, когда он хочет отправить пакет в удаленную сеть через маршрутизатор, подключенный к той же локальной сети, что и данный узел.

Для определения локального адреса по IP-адресу используется протокол разрешения адреса *Address Resolution Protocol, ARP*. Существует также протокол, решающий обратную задачу – нахождение IP-адреса по известному локальному адресу. Он называется реверсивный *ARP – RARP (Reverse Address Resolution Protocol)* и используется при старте бездисковых станций, не знающих в начальный момент своего IP-адреса, но знающих адрес своего сетевого адаптера. В локальных сетях протокол *ARP* использует широковещательные кадры протокола канального уровня для поиска в сети узла с заданным IP-адресом.

Узел, которому нужно выполнить отображение IP-адреса на локальный адрес, формирует *ARP-запрос*, вкладывает его в кадр протокола канального уровня, указывая в нем известный IP-адрес, и рассылает запрос широковещательно. Все узлы локальной сети получают *ARP-запрос* и сравнивают указанный там IP-адрес с собственным. В случае их совпадения узел формирует *ARP-ответ*, в котором указывает свой IP-адрес и свой локальный адрес и отправляет его уже направленно, т. к. в *ARP-запросе* отправитель указывает свой локальный адрес. *ARP-запросы* и *ARP-ответы* используют один и тот же формат пакета. Так как локальные адреса могут в различных типах сетей иметь различную длину, то формат пакета протокола *ARP* зависит от типа сети.

8.6. Система DNS

8.6.1. Символьные имена системы DNS

Для идентификации компьютеров аппаратное и программное обеспечение в сетях *TCP/IP* полагаются на IP-адреса, поэтому для доступа к сетевому ресурсу в параметрах программы вполне достаточно указать IP-адрес, чтобы программа правильно поняла, к какому хосту ей нужно обратиться. Например, команда *ftp://192.45.66.17* будет устанавливать сеанс связи с нужным *ftp-сервером*, а команда *http://203.23.106.33* откроет начальную страницу на корпоративном *web-сервере*. Однако пользователи обычно предпочитают работать с символьными именами компьютеров, и операционные системы локальных сетей приучили их к этому удобному способу. Следовательно, в сетях *TCP/IP* должны существовать символьные имена хостов и механизм для установления соответствия между символьными именами и IP-адресами.

В операционных системах, которые первоначально разрабатывались для работы в локальных сетях, таких как Novell Net Ware, Microsoft Windows или IBMOS/2, пользователи всегда работали с символьными именами компьютеров. Так как локальные сети состояли из небольшого числа компьютеров, то использовались так называемые плоские имена, состоящие из последовательности символов, не разделенных на части. Примерами таких имен являются: *NWL 1*, *mail2*, *MOSCOW SALES 2*. Для установления соответствия между символьными именами и *MAC-адресами* в этих операционных системах применялся механизм ширококвещательных запросов, подобный механизму запросов протокола *ARP*. Так, ширококвещательный способ разрешения имен реализован в протоколе NetBIOS, на котором были построены многие локальные ОС. Так называемые NetBIOS-имена стали на долгие годы одним из основных типов плоских имен в локальных сетях.

Для стека *TCP/IP*, рассчитанного в общем случае на работу в больших территориально распределенных сетях, подобный подход оказывается неэффективным по нескольким причинам.

Для эффективной организации именования компьютеров в больших сетях естественным является применение иерархических составных имен.

В стеке *TCP/IP* применяется доменная система имен, которая имеет иерархическую древовидную структуру, допускающую использование в имени произвольного количества составных частей (рис. 8.4).

Иерархия доменных имен аналогична иерархии имен файлов, принятой во многих популярных файловых системах. Дерево имен начинается с корня, обозначаемого здесь точкой «». Затем следует старшая символьная часть имени, вторая по старшинству символьная часть имени и т. д. Младшая часть имени соответствует конечному узлу сети. В отличие от имен файлов, при записи которых сначала указывается самая старшая составляющая, затем составляющая более низкого уровня и т. д., запись доменного имени начинается с самой младшей составляющей, а заканчивается самой старшей. Составные части доменного имени отделяется друг от друга точкой. Например, в имени *partnering.microsoft.com* составляющая *partnering* является именем одного из компьютеров в домене *microsoft.com*.

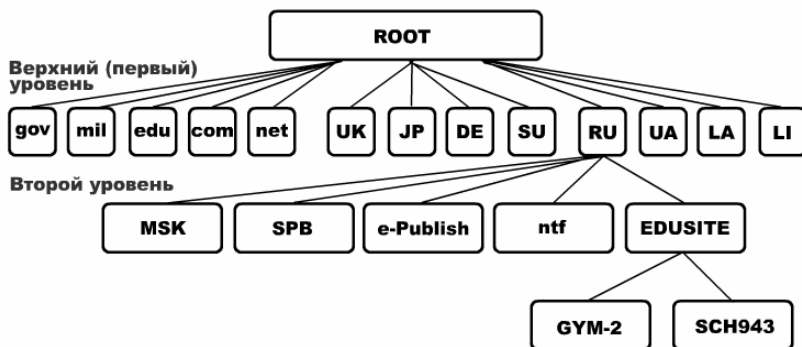


Рис. 8.4. Пространство доменных имен

Разделение имени на части позволяет *разделить административную ответственность* за назначение уникальных имен между различными людьми или

организациями в пределах своего уровня иерархии. Так, для примера, приведенного на рис. 8.5, один человек может нести ответственность за то, чтобы все имена, которые имеют окончание «*ru*», имели уникальную следующую вниз по иерархии часть. Если этот человек справляется со своими обязанностями, то все имена типа *www.ru*, *mail.mmt.ru* или *m2.zil.mmt.ru* будут отличаться второй по старшинству частью.

Разделение административной ответственности позволяет решить проблему образования уникальных имен без взаимных консультаций между организациями, отвечающими за имена одного уровня иерархии. Очевидно, что должна существовать одна организация, отвечающая за назначение имен верхнего уровня иерархии.

Совокупность имен, у которых несколько старших составных частей совпадают, образуют *домен (domain)* имен. Например, имена *wwwl.zil.mmt.ru*, *ftp.zil.mmt.ru*, *yandex.ru* и *sl.mgu.ru* входят в домен *ru*, т. к. все эти имена имеют одну общую старшую часть – имя *ru*. Другим примером является домен *mgu.ru*. Из представленных на рис. 8.5 имен в него входят имена *sl.mgu.ru*, *s2.mgu.ru* и *rn.mgu.ru*. Этот домен образуют имена, у которых две старшие части всегда равны *mgu.ru*. Имя *www.mmt.ru* в домен *mgu.ru* не входит, т. к. имеет отличающуюся составляющую *mtt*.

Если один домен входит в другой домен как его составная часть, то такой домен называют *поддоменом (sub domain)*, хотя название домен за ним также остается. Обычно поддомен называют по имени той его старшей составляющей, которая отличает его от других поддоменов.

Например, поддомен *mtt.ru* обычно называют *поддоменом (доменом) mtt*. Имя *поддомену* назначает администратор вышестоящего домена. Хорошей аналогией домена является каталог файловой системы.

Если в каждом домене и поддомене обеспечивается уникальность имен следующего уровня иерархии, то и вся система имен будет состоять из уникальных имен.

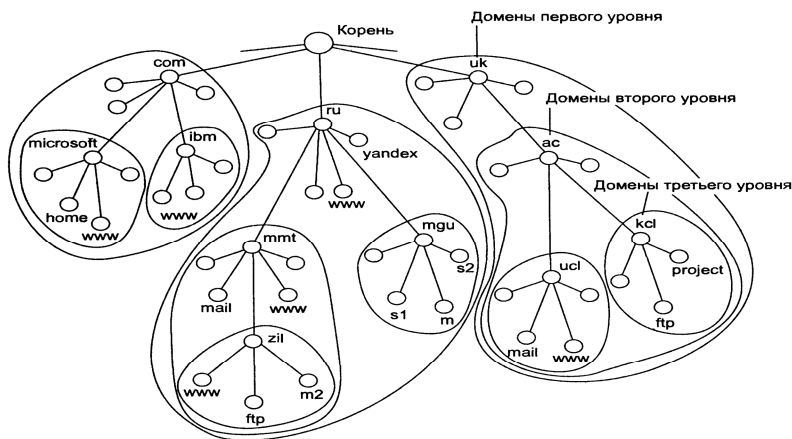


Рис. 8.5. Пример иерархии имен

По аналогии с файловой системой в доменной системе имен различают *краткие имена*, *относительные имена* и *полные доменные имена*. *Краткое имя* – это имя конечного узла сети: хоста или порта маршрутизатора. *Краткое имя* – это лист дерева имен. *Относительное имя* – это составное имя, начинающееся с некоторого уровня иерархии, но не самого верхнего. Например, *www.zil* – это относительное имя. *Полное доменное имя* (*fully qualified domain name, FQDN*) включает составляющие всех уровней иерархии, начиная от краткого имени и кончая корневой точкой: *www.zil.mmt.ru*.

Корневой домен управляется центральными органами Интернета: *LANA* и *iter NIC*. Домены верхнего уровня назначаются для каждой страны, а также на организационной основе. Имена этих доменов должны следовать международному стандарту ISO 3166. Для обозначения стран используются трехбуквенные и двухбуквенные аббревиатуры, например, *ru* (Россия), *uk* (Великобритания), *fin* (Финляндия), *us* (Соединенные Штаты), а для различных типов организаций следующие обозначения:

- *com* – коммерческие организации (например, *microsoft.com*);

- *edu* – образовательные организации (например, *mit.edu*);
- *gov* – правительственные организации (например, *nsf.gov*);
- *org* – некоммерческие организации (например, *fidonet.org*);
- *net* – организации поддержки сетей (например, *nsf.net*).

Каждый домен администрируется отдельной организацией, разбивающая свой домен на поддомены и передающая функции администрирования этих поддоменов другим организациям. Чтобы получить доменное имя, необходимо зарегистрироваться в какой-либо организации, которой организация *Inter NIC* делегировала свои полномочия по распределению имен доменов. В России такой организацией является РосНИИРОС, которая отвечает за делегирование имён поддоменов в домене *ru*. В Казахстане «Казахский центр сетевой информации» – *KazNIC* признан международным регистром Интернет IANA, и является назначенным Менеджером (оператором) по распределению и поддержке доменов Интернет адресного пространства *KZ*.

Компьютеры входят в домен в соответствии со своими составными именами, при этом они могут иметь абсолютно независимые друг от друга IP-адреса, принадлежащие к различным сетям и подсетям. Например, в домен *mgui.ru* могут входить хосты с адресами *132.13.34.15*, *201.22.100.33* *14.0.0.6*.

Доменная система имен реализована в Интернете, но она может работать и как автономная система имен в любой крупной корпоративной сети, которая также использует стек *TCP/IP*, но никак не связана с Интернетом.

8.6.2. Схема работы DNS

В сетях *TCP/IP* соответствие между доменными именами и IP-адресами может устанавливаться средствами: локального хоста; централизованной службы. Вначале развития Интернета на каждом хосте вручную создавался текстовый файл с именем *hosts.txt*, состоящий из некоторого количества строк, каждая содержала одну пару «*доменное имя – IP-адрес*», например: *rhino.acme.com – 102.54.94.97*. С развитием Интернета файлы *hosts.txt* увеличивались в объеме, и создание масштабируемого решения для разрешения имен стало необходимостью.

Для решения этой проблемы была создана специальная служба – *система доменных имен (Domain Name System, DNS)*. *DNS* – это централизованная служба, основанная на распределенной базе отображений «*доменное имя – IP-адрес*». Служба *DNS* использует в своей работе протокол типа «клиент-сервер». В нем определены *DNS-серверы* и *DNS-клиенты*. *DNS-серверы* поддерживают распределенную базу отображений, а *DNS-клиенты* обращаются к серверам с запросами о разрешении доменного имени в IP-адрес.

Служба *DNS* использует текстовые файлы почти такого формата, как и файл *hosts*, и эти файлы администратор также подготавливает вручную. Однако служба *DNS* опирается на иерархию доменов, и каждый сервер службы *DNS* хранит только часть имен сети, а не все имена, как это происходит при использовании файлов *hosts*. С ростом количества узлов в сети проблема масштабирования решается созданием новых доменов и поддоменов имен, добавлением в службу *DNS* новых серверов.

Для каждого домена имен создается свой *DNS-сервер*. Две системы распределения имен на серверах. В первом случае сервер может хранить отображения «*доменное имя – IP-адрес*» для всего домена, включая все его поддомены. Но такое решение плохо масштабируемое, т. к. при добавлении новых поддоменов нагрузка на этот сервер может превысить его возможности. Чаще используется подход, когда сервер домена хранит только имена, которые заканчиваются на следующем ниже уровне иерархии по сравнению с именем домена. (Аналогично каталогу файловой системы, который содержит записи о файлах и подкаталогах, непосредственно в него «входящих».) Именно при такой организации службы *DNS* нагрузка по разрешению имен распределяется более-менее равномерно между всеми *DNS-серверами* сети. Каждый *DNS-сервер*, кроме таблицы отображений имен, содержит ссылки на *DNS-сервер* своих поддоменов. Эти ссылки связывают отдельные *DNS-серверы* в единую службу *DNS*. Ссылки представляют собой IP-адреса соответствующих серверов. Для обслуживания корневого домена выделено несколько дублирующих друг друга *DNS-серверов*, IP-адреса которых являются широко известными.

Процедура разрешения *DNS-имени* во многом аналогична процедуре поиска файловой системой адреса файла по его символьному имени. В обоих случаях составное имя отражает иерархическую структуру организации соответствующих справочников – каталогов файлов или таблиц *DNS*. Здесь домен и доменный *DNS-сервер* являются аналогом каталога файловой системы. Для доменных имен, так же как и для символьных имен файлов, характерна независимость именования от физического местоположения.

Процедура поиска адреса файла по символьному имени заключается в последовательном просмотре каталогов, начиная с корневого. При этом предварительно проверяется кэш и текущий каталог. Для определения IP-адреса по доменному имени также необходимо просмотреть все *DNS-серверы*, обслуживающие цепочку поддоменов, входящих в имя хоста, начиная с корневого домена. Существенным же отличием является то, что файловая система расположена на одном компьютере, а служба *DNS* по своей природе является распределенной.

Две основные схемы разрешения *DNS-имен*. В первом варианте работу по поиску IP-адреса координирует *DNS-клиент*:

- *DNS-клиент* обращается к корневому *DNS-серверу* с указанием полного доменного имени;

- *DNS-сервер* отвечает, указывая адрес следующего *DNS-сервера*, обслуживающего домен верхнего уровня, заданный в старшей части запрошенного имени;

- *DNS-клиент* делает запрос следующего *DNS-сервера*, который отсылает его к *DNS-серверу* нужного поддомена и т. д., пока не будет найден *DNS-сервер*, с соответствующим запрошенному имени IP-адресом. Этот сервер дает окончательный ответ клиенту.

Такая схема взаимодействия называется *нерекурсивной* или *итеративной*, когда клиент сам итеративно выполняет последовательность запросов к разным серверам имен. Так как эта схема загружает клиента достаточно сложной работой, то она применяется редко.

Второй вариант реализует рекурсивную процедуру:

- *DNS-клиент* запрашивает локальный *DNS-сервер*,

обслуживающий поддомен, к которому принадлежит имя клиента;

- если локальный *DNS-сервер* знает ответ, то он возвращает его клиенту; это может соответствовать случаю, когда запрошенное имя входит в тот же поддомен, что и имя клиента, а также случаю, когда сервер уже узнавал данное соответствие для другого клиента и сохранил его в своем КЭШе;

- если локальный сервер не знает ответ, то он выполняет итеративные запросы к корневому серверу и т. д. точно так же, как это делает клиент в первом варианте. Получив ответ, он передает его клиенту, который все это время просто ожидает его от своего локального *DNS-сервера*.

В этой схеме клиент перепоручает работу своему серверу, поэтому схема называется *косвенной*, или *рекурсивной*. Практически все *DNS-клиенты* используют *рекурсивную процедуру*.

Для ускорения поиска IP-адресов *DNS-серверы* широко применяют процедуру кэширования проходящих через них ответов. Чтобы служба *DNS* могла оперативно отрабатывать изменения, происходящие в сети, ответы кэшируются на определенное время – обычно от нескольких часов до нескольких дней.

8.6.3. Динамическое конфигурирование хостов

Для нормальной работы сети каждому сетевому интерфейсу компьютера и маршрутизатора, на которые будут приходить, и с которых будут отправляться IP-пакеты, должен быть назначен IP-адрес. Назначение IP-адресов может происходить вручную в результате выполнения процедуры конфигурирования интерфейса (например, заполнение системы экранных форм). Даже при не очень большом размере сети эта работа представляет для администратора рутинную, а временами и утомительную процедуру. Протокол *Dynamic Host Configuration Protocol (DHCP)* освобождает администратора от этих проблем, автоматизируя процесс назначения IP-адресов.

DHCP автоматизирует процесс конфигурирования сетевых интерфейсов, обеспечивает централизованное управление

распределением адресов и отсутствие их дублирования. Протокол *DHCP* работает в соответствии с моделью *клиент-сервер*. Во время старта системы компьютер, являющийся *DHCP-клиентом*, посылает в сеть широковещательный запрос на получение IP-адреса. *DHCP-сервер* откликается и посылает *сообщение-ответ*, с IP-адресом и другими конфигурационными параметрами. Предполагается, что *DHCP-клиент* и *DHCP-сервер* находятся в одной IP-сети.

DHCP-сервер может работать в режимах:

- ручного назначения статических адресов;
- автоматического назначения статических адресов;
- автоматического распределения динамических адресов.

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

При *ручной* процедуре назначения статических адресов активное участие принимает администратор, который сообщает *DHCP-серверу* информацию о жестком соответствии IP-адресов физическим адресам или другим идентификаторам клиентов. *DHCP-сервер*, пользуясь этой информацией, выдает определенному клиенту один и тот же назначенный ему администратором адрес.

При *автоматическом* назначении *DHCP-сервер* самостоятельно, без вмешательства администратора, выбирает клиенту произвольный IP-адрес из пула наличных IP-адресов. Границы пула задает администратор при конфигурировании *DHCP-сервера*. Адрес дается клиенту из пула в постоянное пользование, то есть с неограниченным сроком аренды. Между идентификатором клиента и его IP-адресом, как и при ручном назначении, существует постоянное соответствие. Оно устанавливается в момент первого назначения *DHCP-сервером* IP-адреса клиенту. При всех последующих запросах сервер возвращает клиенту тот же самый IP-адрес.

При *динамическом* распределении адресов *DHCP-сервер* выдает адрес клиенту на ограниченное время, называемое *временем аренды (lease duration)*, что дает возможность

впоследствии повторно использовать этот IP-адрес для назначения другому компьютеру. Таким образом, помимо основного преимущества *DHCP-автоматизации* рутинной работы администратора по конфигурированию стека TCP/IP на каждом компьютере, динамическое разделение адресов позволяет строить IP-сеть, количество узлов в которой превышает количество имеющихся в распоряжении администратора IP-адресов.

Администратор управляет процессом назначения адресов с помощью параметра *«продолжительность аренды»*, определяющий продолжительность использования назначенного IP-адреса компьютером, перед очередным запросом его от *DHCP-сервера* в аренду.

DHCP-сервер может назначить клиенту не только IP-адрес, но и другие параметры стека *TCP/IP*, необходимые для его эффективной работы, например, такие: маску, IP-адрес маршрутизатора по умолчанию, IP-адрес сервера DNS, доменное имя компьютера и т. п.

8.7. Базовые протоколы TCP/IP

Главную задачу транспортного уровня – передачу данных между прикладными процессами, выполняемые компьютерами в сети, решают протоколы TCP и UDP. Два основных транспортных протокола: протокол управления передачей (TCP) и протокол пользовательских дейтаграмм (UDP). TCP обеспечивает транспортировку данных с установлением соединения, а UDP работает без установления соединения.

Сегменты TCP. Протокол *TCP (Transmission Control Protocol)* обеспечивает надежную транспортировку данных между прикладными процессами путем установления логического соединения.

Информация, поступающая к протоколу *TCP* в рамках логического соединения от протоколов более высокого уровня, рассматривается им как неструктурированный поток байт и буферизуется средствами *TCP*. Для передачи на сетевой уровень из буфера «вырезается» некоторая непрерывная часть данных, называемая *сегментом*.

В протоколе *TCP* предусмотрен случай обращения приложения с запросом о срочной передаче данных (бит *PSH=1*). В этом случае протокол *TCP*, не ожидая заполнения буфера до уровня размеров сегмента, немедленно передает указанные данные в сеть. О таких данных говорят, что они передаются вне потока – *out of band*.

Сегменты, посланные через соединение, могут быть различного размера, участники соединения должны договориться о максимальном размере сегмента, который они будут использовать. Этот размер выбирается таким образом, чтобы при упаковке сегмента в *IP*-пакет он помещался туда целиком, т. е. максимальный размер сегмента не должен превосходить максимального размера поля данных *IP-пакета*. В противном случае необходимо выполнять фрагментацию, т. е. делить сегмент на несколько частей, для того, чтобы он влез в *IP-пакет*.

Таким же образом решается и на сетевом уровне, чтобы избежать фрагментации, выбирается соответствующий максимальный размер *IP-пакета*. Однако при этом должны быть приняты во внимание максимальные размеры поля данных кадров (*MTU*) всех протоколов канального уровня, используемых в сети. Максимальный размер сегмента не должен превышать минимальное значение на множестве всех *MTU* составной сети.

Порты и установление *TCP*-соединений. Пакеты, поступающие на транспортный уровень, организуются операционной системой в виде множества очередей к точкам входа различных прикладных процессов. В терминологии *TCP/IP* такие системные очереди называются *портами*. Адресом назначения, используемый протоколом *TCP* является идентификатор (номер) порта прикладной службы. Номер порта в совокупности с номером сети и номером конечного узла определяют прикладной процесс в сети. Данный набор идентифицирующих параметров называется *сокетом* (*sockets*). Назначение номеров портов прикладным процессам осуществляется:

- централизованно, если процессы популярные общедоступные службы (FTP, telnet и др.);
- локально, для не очень распространенных служб.

Протокол *TCP* ведет для каждого порта две очереди: очередь пакетов, поступающих в данный порт из сети, и очередь пакетов отправляемых данным портом в сеть. Процедура обслуживания протоколом *TCP* запросов, поступающих от нескольких различных прикладных служб, называется *мультиплексированием*. Обратная процедура распределения протоколом *TCP* поступающих от сетевого уровня пакетов между набором высокоуровневых служб, идентифицированных номерами портов, называется *демультиплексированием*. Для организации надежной передачи данных предусматривается установление *логического соединения* между двумя прикладными процессами. В рамках соединения осуществляется обязательное подтверждение правильности приема для всех переданных сообщений, и при необходимости выполняется повторная передача. Соединение в *TCP* позволяет вести передачу данных одновременно в обе стороны, то есть полнодуплексную передачу.

Соединение в протоколе *TCP* идентифицируется парой полных адресов обоих взаимодействующих процессов (оконечных точек). Адрес каждой из оконечных точек включает IP-адрес (номер сети и номер компьютера) и номер порта. Одна оконечная точка может участвовать в нескольких соединениях.

Установление соединения выполняется в следующей последовательности:

1. При установлении соединения одна из сторон является инициатором. Она посылает запрос к протоколу *TCP* на открытие порта для передачи (*active open*);

2. После открытия порта протокол *TCP* на стороне процесса-инициатора посылает запрос процессу, с которым требуется установить соединение;

3. Протокол *TCP* на приемной стороне открывает порт для приема данных (*passive open*) и возвращает квитанцию, подтверждающую прием запроса;

4. Для того чтобы передача могла вестись в обе стороны, протокол на приемной стороне также открывает порт для передачи (*active port*) и также передает запрос к противоположной стороне;

5. Сторона-инициатор открывает порт для приема и возвращает квитанцию. Соединение считается установленным. Далее происходит обмен данными в рамках данного соединения.

Концепция квитирования. В рамках соединения правильность передачи каждого сегмента должна подтверждаться квитанцией получателя. *Квитирование* – это один из традиционных методов обеспечения надежной связи. Идея квитирования состоит в следующем.

Для того чтобы можно было организовать повторную передачу искаженных данных, отправитель нумерует отправляемые единицы передаваемых данных (*кадров*). Для каждого кадра отправитель ожидает от приемника так называемую положительную квитанцию – *служебное сообщение*, извещающее о том, что исходный кадр был получен и данные в нем оказались корректными. Время этого ожидания ограничено – при отправке каждого кадра передатчик запускает таймер, и если по его истечении положительная квитанция не получена, то кадр считается утерянным. В некоторых протоколах приемник, в случае получения кадра с искаженными данными должен отправить отрицательную квитанцию – явное указание того, что данный кадр нужно передать повторно.

Существуют два подхода к организации процесса обмена положительными и отрицательными квитанциями: с простоями и с организацией «окна».

Метод с простоями требует, чтобы источник, пославший кадр, ожидал получения квитанции (положительной или отрицательной) от приемника и только после этого посылал следующий кадр (или повторял искаженный). Из рис. 8.6 видно, что в данном случае производительность обмена данными существенно снижается, хотя передатчик и мог бы послать следующий кадр сразу же после отправки предыдущего, он обязан ждать прихода квитанции. Снижение производительности для этого метода коррекции особенно заметно на низкоскоростных каналах связи, то есть в территориальных сетях.

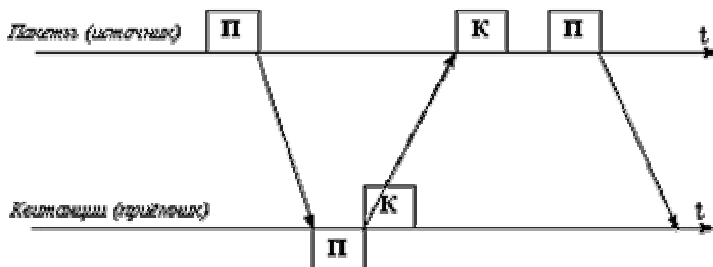


Рис. 8.6. Метод подтверждения корректности передачи кадров с простым источником

Во втором методе для повышения коэффициента использования линии источнику разрешается передать некоторое количество кадров в непрерывном режиме, то есть в максимально возможном для источника темпе, без получения на эти кадры ответных квитанций. Количество кадров, которые разрешается передавать, таким образом, называется *размером окна*. Рис. 8.7 иллюстрирует метод «окна» с размером в W кадров. Обычно кадры при обмене нумеруются циклически, от 1 до W . При отправке кадра с номером 1 источнику разрешается передать еще $W-1$ кадров до получения квитанции на кадр 1. Если же за это время квитанция на кадр 1 так и не пришла, то процесс передачи приостанавливается, и по истечению некоторого тайм-аута кадр 1 считается утерянным (или квитанция на него утеряна) и он передается снова.

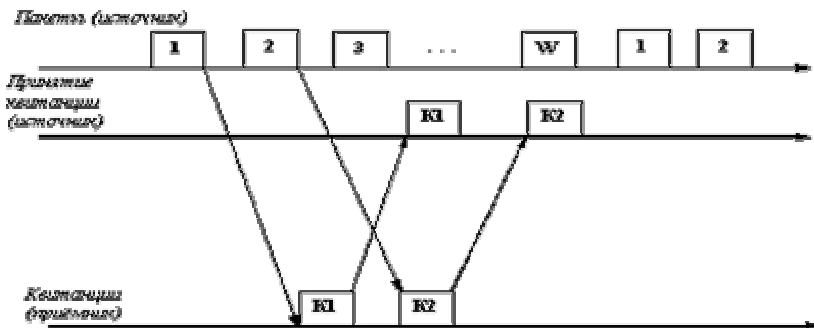


Рис. 8.7. Метод «окна» – непрерывная отправка пакетов

Если же поток квитанций поступает более-менее регулярно, в пределах допуска в W кадров, то скорость обмена достигает максимально возможной величины для данного канала и принятого протокола. Этот алгоритм называют алгоритмом «скользящего окна». При каждом получении квитанции окно перемещается (*скользит*), захватывая новые данные, которые разрешается передавать без подтверждения.

Реализация скользящего окна в протоколе TSP. В протоколе TSP реализована разновидность алгоритма квитирования с использованием окна. Особенность этого алгоритма состоит в том, что, хотя единицей передаваемых данных является сегмент, окно определено на множестве нумерованных байт неструктурированного потока данных, поступающих с верхнего уровня и буферизируемых протоколом TSP.

Квитанция посылается только в случае правильного приема данных, отрицательные квитанции не посылаются. Таким образом, отсутствие квитанции означает либо прием искаженного сегмента, либо потерю сегмента, либо потерю квитанции.

В качестве квитанции получатель сегмента отправляет ответное сообщение (сегмент), которое помещает число, на единицу превышающее максимальный номер байта в полученном сегменте. Если размер окна равен W , а последняя квитанция содержала значение N , то отправитель может посылать новые сегменты до тех пор, пока в очередной сегмент не попадет байт с номером $N+W$. Этот сегмент выходит за рамки окна, и передачу в таком случае необходимо приостановить до прихода следующей квитанции.

Выбор тайм-аута. Выбор времени ожидания (тайм-аута) очередной квитанции является важной задачей, результат решения которой влияет на производительность протокола TSP.

Тайм-аут не должен быть слишком коротким, чтобы по возможности исключить избыточные повторные передачи, снижающие полезную пропускную способность системы. Но он не должен быть и слишком большим, чтобы избежать длительных простоев, связанных с ожиданием несуществующей или «заблудившейся» квитанции.

При выборе величины тайм-аута должны учитываться скорость и надежность физических линий связи, их протяженность и многие другие подобные факторы. В протоколе *TCP* тайм-аут определяется с помощью достаточно сложного адаптивного алгоритма, идея которого состоит в следующем. При каждой передаче засекается время от момента отправки сегмента до прихода квитанции о его приеме (время оборота). Получаемые значения времен оборота усредняются с весовыми коэффициентами, возрастающими от предыдущего замера к последующему. Это делается с тем, чтобы усилить влияние последних замеров. В качестве тайм-аута выбирается среднее время оборота, умноженное на некоторый коэффициент. Практика показывает, что значение этого коэффициента должно превышать 2. В сетях с большим разбросом времени оборота при выборе тайм-аута учитывается и дисперсия этой величины.

Реакция на перегрузку сети. Варьируя величиной окна, можно повлиять на загрузку сети. Чем больше окно, тем большую порцию неподтвержденных данных можно послать в сеть. Если сеть не справляется с нагрузкой, то возникают очереди в промежуточных узлах – маршрутизаторах и в конечных узлах – компьютерах.

При переполнении приемного буфера конечного узла «перегруженный» протокол *TCP*, отправляя квитанцию, помещает в нее новый, уменьшенный размер окна. Если он совсем отказывается от приема, то в квитанции указывается окно нулевого размера. Однако даже после этого приложение может послать сообщение на отказавшийся от приема порт. Для этого, сообщение должно сопровождаться пометкой «срочно» (бит «*URG*» =1 в запросе). В такой ситуации порт обязан принять сегмент, даже если для этого придется вытеснить из буфера уже находящиеся там данные.

После приема квитанции с нулевым значением окна протокол-отправитель время от времени делает контрольные попытки продолжить обмен данными. Если протокол-приемник уже готов принимать информацию, то в ответ на контрольный запрос он посылает квитанцию с указанием ненулевого размера окна.

Другим проявлением перегрузки сети является переполнение буферов в маршрутизаторах. В таких случаях они могут централизованно изменить размер окна, посылая управляющие сообщения некоторым конечным узлам, что позволяет им дифференцированно управлять интенсивностью потока данных в разных частях сети.

Формат сообщений TCP. Сообщения протокола TCP называются сегментами и состоят из заголовка и блока данных. Заголовок сегмента имеет следующие поля (рис. 8.8):

Порт отправителя (16 бит)								Порт получателя (16 бит)							
Позиция сегмента (32 бита) – последовательный номер															
Первый ожидаемый байт (32 бита) – номер подтверждения															
Длина заголовка (4 бита)	Резерв (6 бит)	U	A	P	R	S	F	Размер окна (16 бит)							
		R	C	S	S	Y	I								
		G	K	H	T	N	N								
Контрольная сумма сегмента (16 бит)								Указатель срочных данных (16 бит)							
Опции и заполнитель															
Данные															

Рис. 8.8. Структура заголовка сегмента и формат пакета TCP

Поле «Порт источника» (*SOURCE PORT*) занимает 2 байта, идентифицирует процесс-отправитель; Поле «Порт назначения» (*DESTINATION PORT*) занимает 2 байта, идентифицирует процесс-получатель; Поле «Последовательный номер» (*SEQUENCE NUMBER*) занимает 4 байта, указывает номер байта, который определяет смещение сегмента относительно потока отправляемых данных; Поле «Подтвержденный номер» (*ACKNOWLEDGEMENT NUMBER*) занимает 4 байта, содержит максимальный номер байта в полученном сегменте, увеличенный на 1, это значение используется в качестве квитанции; Поле «Длина заголовка» (*HLLEN*) занимает 4 бита, указывает длину заголовка сегмента TCP, измеренную в 32-битовых словах. Длина заголовка не фиксирована и может изменяться в зависимости от значений, устанавливаемых в поле «Опции»; Поле «Резерв» (*RESERVED*) занимает 6 битов, поле зарезервировано для последующего использования; Поле

«Кодовые биты» (*CODE BITS*) занимают 6 битов, содержат служебную информацию о типе данного сегмента, задаваемую установкой в единицу соответствующих бит этого поля: «*URG*» – срочное сообщение; «*ACK*» – квитанция на принятый сегмент; «*PSH*» – запрос на отправку сообщения без ожидания заполнения буфера; «*RST*» – запрос на восстановление соединения; «*SYN*» – сообщение, используемое для синхронизации счетчиков переданных данных при установлении соединения; Поле «*FIN*» – признак достижения передающей стороной последнего байта в потоке передаваемых данных; Поле «Окно» (*WINDOW*) занимает 2 байта, содержит объявляемое значение размера окна в байтах; Поле «Контрольная сумма» (*CHECKSUM*) занимает 2 байта, рассчитывается по сегменту; Поле «Указатель срочности» (*URGENT POINTER*) занимает 2 байта, используется совместно с кодовым битом *URG*, указывает на конец данных, которые необходимо срочно принять, несмотря на переполнение буфера; Поле «Опции» (*OPTIONS*) имеет переменную длину и может вообще отсутствовать, максимальная величина поля 3 байта; используется для решения вспомогательных задач, например, при выборе максимального размера сегмента; Поле «Заполнитель» (*PADDING*) может иметь переменную длину, представляет собой фиктивное поле, используемое для доведения размера заголовка до целого числа 32-битовых слов.

Протокол дейтаграмм пользователя (*UDP*). *UDP (User Data Protocol)* – транспортный протокол без соединений. Задача этого протокола – передача данных без гарантий доставки. Поэтому его пакеты могут быть потеряны, продублированы или прийти не в том порядке, в котором они были отправлены. В вышележащий уровень данные передаются в не отсортированном виде. Не предусмотрено исправление ошибок. Протокол *UDP* намного проще, чем *TCP*: он полезен в ситуациях, когда мощные механизмы обеспечения надежности протокола *TCP* не обязательны. Заголовок *UDP* имеет всего четыре поля (рис. 8.9).

Порт отправителя (16 бит)	Порт получателя (16 бит)
Длина дейтаграммы (16 бит)	Контрольная сумма (16 бит) дейтаграммы
Данные	

Рис. 8.9. Структура заголовка дейтаграммы и формат пакета UDP

Поля «*Порт отправителя*» (*source port*, 16 бит) и «*Порт получателя*» (*destination port*, 16 бит) содержат номера портов отправителя и получателя. Поле «*Длина дейтаграммы*» (*length*, 16 бит). Содержит полную длину дейтаграммы в байтах. Поле «*Контрольная сумма дейтаграммы*» (*checksum UDP*, 16 бит) обеспечивает проверку целостности пакета. При обнаружении ошибки дейтаграмма отбрасывается. Заполнение поля «*Контрольная сумма дейтаграммы*» нулями понимается как отказ от расчета контрольной суммы. Для того чтобы этот отказ отличался от того (редкого, но возможного) случая, когда рассчитанная контрольная сумма равна нулю, все биты поля контрольной суммы устанавливаются в состояние 1.

Выводы:

Стек TCP/IP используются три типа адресов: локальные (аппаратные), IP- адреса и символьные доменные имена. Все адреса присваиваются узлам составной сети независимо друг от друга. IP-адрес имеет длину 4 байта и состоит из номера сети и номера узла. Для определения границы, отделяющей номер сети от номера узла, сегодня используются два подхода. Первый основан на применении классов адресов, второй – масок.

Номер сети назначается централизованно, если сеть является частью Интернета. Назначение IP-адресов узлам сети может происходить либо вручную (администратор сам ведет списки свободных и занятых адресов и конфигурирует сетевой интерфейс), либо автоматически (с использованием протокола DHCP).

Протокол разрешения адресов (ARP) устанавливает соответствие между IP-адресом и аппаратным адресом сетевого интерфейса.

Стек TCP/IP применяет систему доменных символьных имен, имеющую иерархическую древовидную структуру, допускающую использование в имени произвольного количества составных частей. Совокупность имен, с совпадающими несколькими старшими составными частями, образуют домен имен. Доменные имена назначаются централизованно, если сеть является частью Интернета, в противном случае – локально.

Соответствие между доменными именами и IP-адресами может устанавливаться как средствами локального хоста с использованием файла hosts, так и с помощью централизованной службы DNS.

Основной задачей протоколов TCP и UDP является передача данных между прикладными процессами различных конечных узлов сети. Дейтаграммный протокол UDP работает без установления логического соединения, не гарантирует доставку своих сообщений, не компенсирует ненадежность дейтаграммного протокола IP.

Системные очереди к точкам входа прикладных процессов называются портами, которые идентифицируются номерами и однозначно определяют приложение в пределах компьютера. Процессы популярных общедоступных служб (FTP, telnet, HTTP, TFTP, DNS и т. п.), используют централизованно закрепленными стандартными номерами.

TCP выполняет надежный обмен данными путем установления логических соединений, однозначно идентифицируемых парой сокетов. Сокетом прикладного процесса называется пара из IP-адреса и номер порта. Для управления потоком в TCP-соединения используется специфический вариант алгоритма «скользящего окна». Сторона-получатель передает стороне-отправителю размер окна приема в байтах.

Контрольные вопросы:

1. Какие существуют типы адресов стека TCP/IP?
2. Форматы и классы IP-адресов.
3. Как применяются маски, и для каких целей?
4. Виды отображения IP-адресов.
5. Каков порядок распределения IP-адресов?

6. Назначение доменных имен и их организация.
7. Протокол надежной доставки сообщений ТСП. Понятие сегменты ТСП?
8. Каким образом назначаются порты, устанавливается ТСП-соединение?
9. В чем заключается концепция квитиования?
10. Как реализуется скользящее окно в протоколе ТСП?

9. ПРОТОКОЛ МЕЖСЕТЕВОГО ВЗАИМОДЕЙСТВИЯ

9.1. Формат IP-пакета

Основным протоколом межсетевого уровня является межсетевой протокол (*Internet Protocol, IP*). В его задачу входит продвижение пакета между подсетями – от одного пограничного маршрутизатора до другого, до тех пор пока пакет не попадет в сеть назначения. *Internet Protocol* относится к протоколам без установления соединения, поддерживает обработку каждого IP-пакета как независимой единицы обмена, не связанной с другими IP-пакетами. Протокол не содержит в себе механизмов, обеспечивающих достоверность конечных данных. При возникновении какой либо ошибки при продвижении пакета протокол не предпринимает каких либо действий для ее исправления.

К основным функциям протокола IP относятся:

- перенос между сетями различных типов адресной информации в унифицированной форме;
- сборка и разборка пакетов при передаче их между сетями с различным максимальным значением длины пакета.

Единица передаваемых данных в IP протоколе называется IP-дейтаграммой (*datagramm*) или IP-пакетом. Применяются два типа IP протокола (IPv4 и IPv6): протокол IPv4 с 32 разрядным и протокол IPv6 со 128 разрядным адресом.

Протокол IPv4. На рис. 9.1 представлен формат IP-пакета. IPv4-заголовок составлен из 32 разрядных 5 или 6 слов (т. е. 20 или 24 байта).

16										17										32									
Версия		Длина заголовка		Тип сервиса						Длина дейтаграммы																			
				PR	D	T	R																						
Идентификатор											DF	MF	Смещение фрагмента																
Время жизни				Протокол						Контрольная сумма заголовка																			
IP-Адрес отправителя																													
IP-Адрес получателя																													
Резерв (опции и заполнитель)																													
ДАННЫЕ																													

Рис.9.1. Формат дейтаграммы IPv4

Заголовок пакета имеет следующие поля:

Поле «*Номер версии*» (VERS) указывает версию протокола IP. Сейчас повсеместно используется версия 4 и готовится переход на версию 6, называемую также *IPng* (*IP next generation*). Поле «*Длина заголовка*» (HLEN) занимает 4 бита и указывает значение длины заголовка, измеряемое в 32-битовых словах. Обычно заголовок имеет длину в 20 байт (пять 32-битовых слов), но при увеличении объема служебной информации эта длина может быть увеличена за счет использования дополнительных байтов в поле «*Резерв*» (*IP OPTIONS*). Поле «*Тип сервиса*» (*SERVICE TYPE*) занимает 1 байт и задает приоритетность пакета и вид критерия выбора маршрута. Первые 3 бита этого поля образуют подполе «*Приоритет пакета*» (*PRECEDENCE*). Приоритет может иметь значения от 0 (нормальный пакет) до 7 (пакет управляющей информации). Маршрутизаторы и компьютеры могут принимать во внимание приоритет пакета и обрабатывать более важные пакеты в первую очередь. Поле «*Тип сервиса*» содержит также 3 бита, определяющих критерий выбора маршрута. Установленный бит «*D*» (*delay*) говорит о выборе маршрута для минимизации задержки доставки данного пакета, бит «*T*» – для максимизации пропускной способности, а бит «*R*» – для максимизации надежности доставки. Поле «*Общая длина*» (*TOTAL LENGTH*) занимает 2 байта и указывает общую длину пакета с учетом заголовка и поля данных. Поле «*Идентификатор пакета*» (*IDENTIFICATION*) занимает 2 байта и используется для распознавания пакетов, образовавшихся путем фрагментации исходного пакета. Все фрагменты должны иметь одинаковое значение этого поля. Поле «*Флаги*» (*FLAGS*) занимает 3 бита, указывает на возможность фрагментации пакета (установленный бит «*Do not Fragment – DF*» –запрещает фрагментировать данный пакет), а также на принадлежность пакета промежуточному или последнему фрагменту исходного пакета (установленный бит «*More Fragments – MF*» – говорит, о промежуточном фрагменте). Поле «*Смещение фрагмента*» (*FRAGMENTOFFSET*) занимает 13 бит, указывает в байтах смещение поля данных пакета от начала общего поля исходного пакета, подвергнутого фрагментации. Используется при сборке/разборке фрагментов пакетов при их передачах между сетями с различными величинами

максимальной длины пакета. Поле *«Время жизни» (TIME TO LIVE)* занимает 1 байт и указывает предельный срок, в течение которого пакет может перемещаться по сети. Время жизни данного пакета измеряется в секундах и задается источником передачи средствами протокола IP. На шлюзах и в других узлах сети по истечении каждой секунды из текущего времени жизни вычитается 1; вычитается также при каждой транзитной передаче (даже если не прошла секунда). При истечении времени жизни пакет аннулируется. *«Идентификатор»* протокола верхнего уровня (*PROTOCOL*) занимает 1 байт и указывает, какому протоколу верхнего уровня принадлежит пакет (протоколы TCP, UDP или RIP). *«Контрольная сумма» (HEADER CHECKSUM)* занимает 2 байта, она рассчитывается по всему заголовку. Поля *«Адрес источника» (SOURCE IP ADDRESS)* и *«Адрес назначения» (DESTINATION IP ADDRESS)* имеют одинаковую длину – 32 бита, и одинаковую структуру. Поле *«Резерв» (IPTIONS)* является необязательным и используется обычно при отладке сети. Это поле состоит из нескольких подполей, каждое из которых может быть 1 из 8 predetermined типов. В этих подполях можно указывать точный маршрут прохождения маршрутизаторов, регистрировать проходимые пакетом маршрутизаторы, помещать данные системы безопасности, а также временные отметки. Так как число подполей может быть произвольным, то в конце поля *«Резерв»* должно быть добавлено несколько байт для выравнивания заголовка пакета по 32-битной границе. Максимальная длина поля *«Данные»* пакета ограничена разрядностью поля, определяющего эту величину, и составляет 65535 байтов, но при передаче по сетям различного типа длина пакета выбирается с учетом максимальной длины пакета протокола нижнего уровня, несущего IP-пакеты. Если это кадры Ethernet, то выбираются пакеты с максимальной длиной в 1500 байтов, уместяющиеся в поле данных кадра Ethernet.

Разбиение информации на пакеты не дает возможности монополизации сети одним пользователем. В отличие от протоколов прикладного и основного уровней протокол IP устанавливается не только на хостах, но и на всех шлюзах (рис. 9.2).

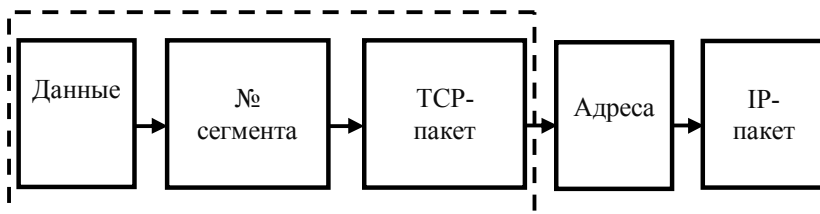


Рис. 9.2. Формирование IP - пакета

9.2. Пример взаимодействия узлов с использованием протокола IP

Рассмотрим пример интерсети, приведенной на рис. 9.3, как происходит взаимодействие компьютеров через маршрутизаторы и доставка пакетов компьютеру назначения.

Пусть в примере пользователь компьютера *cit.dol.ru*, находящийся в сети Ethernet с IP-адресом *194.87.23.0* (адрес класса C), хочет взаимодействовать по протоколу FTP с компьютером *sl.msk.su*, принадлежащем сети Ethernet с IP-адресом *142.06.0.0* (адрес класса B). Компьютер *cit.dol.ru* имеет IP-адрес *194.87.23.1.17*, а компьютер *sl.msk.su* - IP-адрес *142.06.13.14*.

1. Пользователь компьютера *cit.dol.ru* знает символьное имя компьютера *sl.msk.su*, но не знает его IP-адреса, поэтому он набирает команду: *>ftpsl.msk.su* для организации *ftp-сеанса*.

2. В компьютере *cit.dol.ru* должны быть заданы некоторые параметры для стека *TCP/IP*, чтобы он мог выполнить поставленную перед ним задачу.

В число этих параметров должны входить собственный IP-адрес, IP-адрес DNS-сервера и IP-адрес маршрутизатора по умолчанию. Так как к сети Ethernet, к которой относится компьютер *cit.dol.ru*, подключен только один маршрутизатор, то таблица маршрутизации конечным узлам этой сети не нужна, достаточно знать IP-адрес маршрутизатора по умолчанию. В данном примере он равен *194.87.23.1*.

Так как пользователь в команде *ftp* не указал IP-адрес узла, с которым он хочет взаимодействовать, то стек *TCP/IP* должен

определить его самостоятельно. Он может сделать запрос к серверу *DNS* по имеющемуся у него IP-адресу, но сначала компьютер просматривает свою собственную таблицу соответствия символьных имен и IP-адресов. Такая таблица хранится в виде текстового файла простой структуры – каждая его строка содержит запись об одном символьном имени и его IP-адресе. В ОС Unix такой файл традиционно носит имя *HOSTS*.

1. Считаем, что компьютер *cit.dol.ru* имеет файл *HOSTS*, а в нем есть строка *142.06.13.14 sl.msk.su*. Поэтому разрешение имени выполняется локально, так что протокол IP может теперь формировать IP-пакеты с адресом назначения *142.06.13.14* для взаимодействия с компьютером *sl.msk.su*.

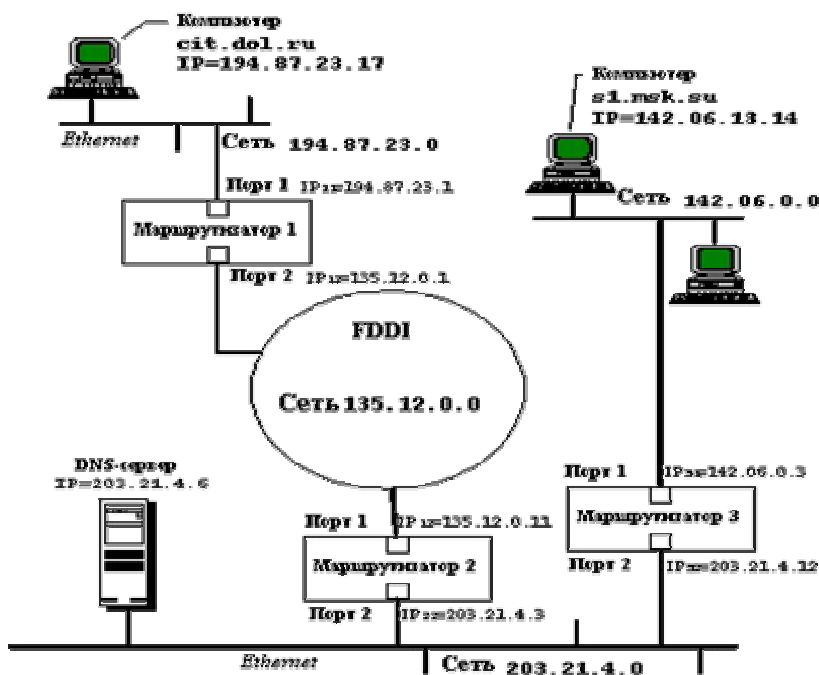


Рис. 9.3. Пример взаимодействия компьютеров через интернет

2. Протокол IP компьютера *cit.dol.ru* проверяет, нужно ли маршрутизировать пакеты для адреса *142.06.13.14*. Так как адрес сети назначения равен *142.06.0.0*, а адрес сети, к которой принадлежит компьютер, равен *194.87.23.0*, то маршрутизация необходима.

3. Компьютер *cit.dol.ru* начинает формировать кадр Ethernet для отправки IP-пакета маршрутизатору по умолчанию с IP-адресом *194.87.23.1*. Для этого ему нужен *MAC-адрес* порта маршрутизатора, подключенного к его сети. Этот адрес, уже находится в кэш-таблице протокола *ARP* компьютера, если он хотя бы раз за последнее включение обменивался данными с компьютерами других сетей. Пусть адрес в нашем примере был найден именно в кэш-памяти. Обозначим его *MAC₁₁*, в соответствии с номером маршрутизатора и его порта.

4. В результате компьютер *cit.dol.ru* отправляет по локальной сети кадр Ethernet, имеющий следующие поля:

<i>DA (Ethernet)</i>	<i>...</i>	<i>DESTINATION IP</i>	<i>...</i>
<i>MAC₁₁</i>		<i>142.06.13.14</i>	

5. Кадр принимается портом 1 маршрутизатора 1 в соответствии с протоколом Ethernet, так как *MAC-узел* этого порта распознает свой адрес *MAC₁₁*. Протокол Ethernet извлекает из этого кадра IP-пакет и передает его программному обеспечению маршрутизатора, реализующему протокол IP. Протокол IP извлекает из пакета адрес назначения и просматривает записи своей таблицы маршрутизации. Пусть маршрутизатор 1 имеет в своей таблице маршрутизации запись *142.06.0.0/24 135.12.0.11/24 2 1*, которая говорит о том, что пакеты для сети *142.06.0.0/24* нужно передавать маршрутизатору *135.12.0.11*, подключенному к той же сети, что и порт 2 маршрутизатора 1.

6. Маршрутизатор 1 просматривает параметры порта 2 и находит, что он подключен к сети FDDI. Так как сеть FDDI имеет значение максимального транспортируемого блока *MTU* больше, чем сеть Ethernet, то фрагментация поля данных IP-пакета не требуется. Поэтому маршрутизатор 1 формирует кадр

формата FDDI, в котором указывает *MAC-адрес* порта маршрутизатора 2, который он находит в своей кэш-таблице протокола *ARP*:

<i>DA (FDDI)</i>	<i>...</i>	<i>DESTINATION IP</i>	<i>...</i>
<i>MAC₂₁</i>		<i>142.06.13.14</i>	

7. Аналогично действует маршрутизатор 2, формируя кадр Ethernet для передачи пакета маршрутизатору 3 по сети Ethernet с IP-адресом *203.21.4.0*:

<i>DA (Ethernet)</i>	<i>...</i>	<i>DESTINATION IP</i>	<i>...</i>
<i>MAC₃₂</i>		<i>142.06.13.14</i>	

8. После поступления пакета в маршрутизатор сети назначения - маршрутизатор 3, появляется возможность передачи этого пакета компьютеру назначения. Маршрутизатор 3 видит, что пакет нужно передать в сеть *142.06.0.0*, которая непосредственно подключена к его первому порту. Поэтому он посылает *ARP-запрос* по сети Ethernet с IP-адресом компьютера *sl.msk.su* (считаем, что этой информации в его кэше нет), получает ответ, содержащий адрес *MAC_{sl}*, и формирует кадр Ethernet, доставляющий IP-пакет по локальной сети адресату.

<i>DA (Ethernet)</i>	<i>...</i>	<i>DESTINATION IP</i>	<i>...</i>
<i>MAC_{sl}</i>		<i>142.06.13.14</i>	

9.3. Управление фрагментацией

Протоколы транспортного уровня (протоколы TCP или UDP), пользующиеся сетевым уровнем для отправки пакетов, считают, что максимальный размер поля данных IP-пакета равен 65535, и поэтому могут передать ему сообщение такой длины для транспортировки через интернет. В функции уровня IP входит разбиение слишком длинного для конкретного типа составляющей сети сообщения на более короткие пакеты с

созданием соответствующих служебных полей, нужных для последующей сборки фрагментов в исходное сообщение.

В большинстве типов локальных и глобальных сетей определяется такое понятие как максимальный размер поля данных кадра или пакета, в которые должен *инкапсулировать* свой пакет протокол IP. Эту величину обычно называют максимальной единицей транспортировки – *Maximum Transfer Unit, MTU*. Сети Ethernet имеют значение *MTU*, равное 1500 байт, сети FDDI – 4096 байт, а сети X.25 чаще всего работают с *MTU* в 128 байт.

Работа протокола IP по фрагментации пакетов при передаче их между сетями с разными максимальными размерами, в хостах и маршрутизаторах иллюстрируется рис. 9.4. Обозначения K1 и Ф1 – канальный и физический уровень сети 1, K2 и Ф2 – канальный и физический уровень сети 2.

Пусть компьютер 1 связан с сетью, имеющей значение *MTU* в 4096 байтов, например, с сетью FDDI. При поступлении на IP-уровень компьютера 1 сообщения от транспортного уровня размером в 5600 байтов, протокол IP делит его на два IP-пакета, устанавливая в первом пакете *признак фрагментации* и присваивая пакету *уникальный идентификатор*, например 486. В первом пакете величина поля смещения равна 0, а во втором – 2800. Признак фрагментации во втором пакете равен нулю, что показывает, что это последний фрагмент пакета. Общая величина IP-пакета составляет 2800+20 (размер заголовка IP), т. е. 2820 байтов, что умещается в поле данных кадра FDDI.

Далее компьютер 1 передает эти пакеты на канальный уровень K1, а затем и на физический уровень Ф1, который отправляет их маршрутизатору, связанному с данной сетью.

Маршрутизатор видит по сетевому адресу, что прибывшие два пакета нужно передать в сеть 2, которая имеет меньшее значение *MTU*, равное 1500. Вероятно, это сеть Ethernet. Маршрутизатор извлекает фрагмент транспортного сообщения из каждого пакета FDDI и делит его еще пополам, чтобы каждая часть уместилась в поле данных кадра Ethernet. Затем он формирует новые пакеты IP, каждый из которых имеет длину 1400+20=1420 байтов, что меньше 1500 байтов, поэтому они нормально помещаются в поле данных кадров Ethernet.

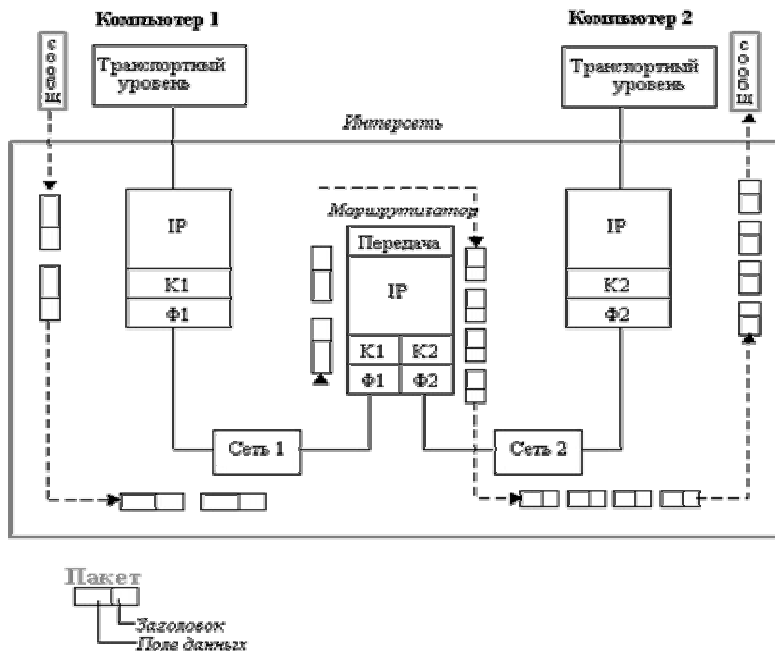


Рис. 9.4. Фрагментация IP-пакетов

В результате в компьютер 2 по сети Ethernet приходит четыре IP-пакета с общим идентификатором 486, что позволяет протоколу IP, работающему в компьютере 2, правильно собрать исходное сообщение. Если пакеты пришли не в том порядке, в котором были посланы, то смещение укажет правильный порядок их объединения.

IP-маршрутизаторы не собирают фрагменты пакетов в более крупные пакеты, даже если на пути встречается сеть, допускающая такое укрупнение. Это связано с тем, что отдельные фрагменты сообщения могут перемещаться по интернету по различным маршрутам, поэтому нет гарантии, что все фрагменты проходят через какой-либо промежуточный маршрутизатор на их пути.

При приходе первого фрагмента пакета узел назначения запускает таймер, который определяет максимально допустимое

время ожидания прихода остальных фрагментов этого пакета. Если таймер истекает раньше прибытия последнего фрагмента, то все полученные к этому моменту фрагменты пакета отбрасываются, а в узел, пославший исходный пакет, направляется сообщение об ошибке с помощью протокола ICMP.

9.4. Маршрутизация с помощью IP-адресов

Рассмотрим теперь принципы, на основании которых в сетях IP происходит выбор маршрута передачи пакета между сетями.

В выборе маршрута принимают участие и маршрутизаторы и конечные узлы – компьютеры. Пример, приведенный на рис. 9.5, демонстрирует эту необходимость. Здесь в локальной сети имеется несколько маршрутизаторов, и компьютер должен выбирать, какому из них следует отправить пакет.



Рис. 9.5. Выбор маршрутизатора конечным узлом

Длина маршрута может существенно измениться в зависимости от того, какой маршрутизатор выберет компьютер для передачи своего пакета на сервер, расположенный, например, в Германии, если маршрутизатор 1 соединен выделенной линией с маршрутизатором в Копенгагене, а маршрутизатор 2 имеет спутниковый канал, соединяющий его с Токио.

В стеке *TCP/IP* маршрутизаторы и конечные узлы

принимают решения о том, кому передавать пакет для его успешной доставки узлу назначения, на основании так называемых таблиц маршрутизации (*routing tables*).

Следующая таблица 9.1 представляет собой типичный пример таблицы маршрутов, использующей IP-адреса сетей.

В этой таблице в столбце «Адрес сети назначения» указываются адреса всех сетей, которым данный маршрутизатор может передавать пакеты.

Таблица 9.1.

Пример таблицы маршрутизации

Адрес сети назначения	Адрес следующего маршрутизатора	Номер выходного порта	Расстояние до сети назначения
56.0.0.0	198.21.17.7	1	20
56.0.0.0	213.34.12.4	2	130
116.0.0.0	213.34.12.4	2	1450
129.13.0.0	198.21.17.6	1	50
198.21.17.0	-	2	0
213.34.12.0	-	1	0
default	198.21.17.7	1	-

В стеке *TCP/IP* принят так называемый *одношаговый подход* к оптимизации маршрута продвижения пакета (*next-hop routing*) – каждый маршрутизатор и конечный узел принимает участие в выборе только одного шага передачи пакета. Поэтому в каждой строке таблицы маршрутизации указывается не весь маршрут в виде последовательности IP-адресов маршрутизаторов, через которые должен пройти пакет, а только один IP-адрес – адрес следующего маршрутизатора, которому нужно передать пакет. Вместе с пакетом следующему маршрутизатору передается ответственность за выбор следующего шага маршрутизации. Одношаговый подход к маршрутизации снимает ограничение на максимальное количество транзитных маршрутизаторов на пути пакета.

Маршрутизация от источника – *Source Routing* предполагает выбор маршрута конечным узлом или первым

маршрутизатором на пути пакета, все остальные маршрутизаторы обрабатывают выбранный маршрут, осуществляя коммутацию пакетов, т. е. передачу их с одного порта на другой. Алгоритм *Source Routing* применяется в сетях IP только для отладки, когда маршрут задается в поле «Резерв пакета» (*IP OPTIONS*).

В случае если в таблице маршрутов имеется более одной строки, соответствующей одному и тому же адресу сети назначения, то при принятии решения о передаче пакета используется та строка, в которой указано наименьшее значение в поле «Расстояние до сети назначения».

При этом под расстоянием понимается любая метрика, используемая в соответствии с заданным в сетевом пакете классом сервиса. Это может быть количество транзитных маршрутизаторов в данном маршруте (количество хопов от *hop* – прыжок), время прохождения пакета по линиям связи, надежность линий связи, или другая величина, отражающая качество данного маршрута по отношению к конкретному классу сервиса (рис. 9.6).

Если маршрутизатор поддерживает несколько классов сервиса пакетов, то таблица маршрутов составляется и применяется отдельно для каждого вида сервиса критерия выбора маршрута.

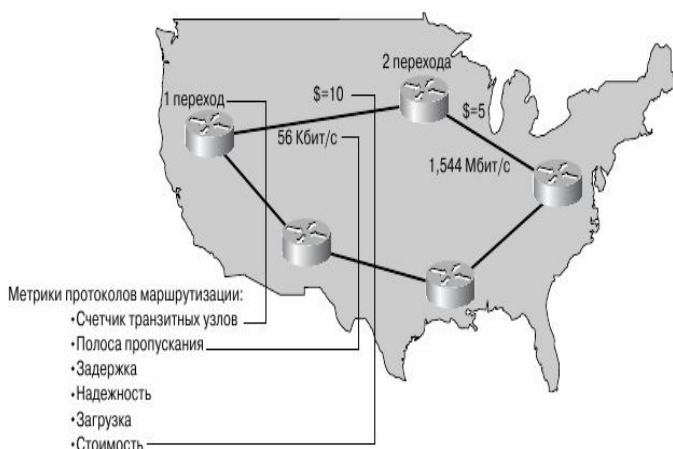


Рис. 9.6. Метрики маршрутизации

Для отправки пакета следующему маршрутизатору требуется знание его локального адреса, но в стеке *TCP/IP* в таблицах маршрутизации принято использование только IP-адресов для сохранения их универсального формата, не зависящего от типа сетей, входящих в интернет. Для нахождения локального адреса по известному IP-адресу необходимо воспользоваться протоколом *ARP*.

Конечный узел, как и маршрутизатор, имеет в своем распоряжении таблицу маршрутов унифицированного формата и на основании ее данных принимает решение, какому маршрутизатору нужно передавать пакет для сети *N*. Решение о том, что этот пакет нужно вообще маршрутизировать, компьютер принимает в том случае, когда он видит, что адрес сети назначения пакета отличается от адреса его собственной сети (каждому компьютеру при конфигурировании администратор присваивает его IP-адрес или несколько IP-адресов, если компьютер одновременно подключен к нескольким сетям). Когда компьютер выбрал следующий маршрутизатор, то он просматривают кэш-таблицу адресов своего протокола *ARP* и, может быть, находит там соответствие IP-адреса следующего маршрутизатора его *MAC-адресу*. Если же нет, то по локальной сети передается широковещательный *ARP-запрос* и локальный адрес извлекается из *ARP-ответа*.

После этого компьютер формирует кадр протокола, используемого на выбранном порту, например, кадр Ethernet, в который помещает *MAC-адрес* маршрутизатора. Маршрутизатор принимает кадр Ethernet, извлекает из него пакет IP и просматривает свою таблицу маршрутизации для нахождения следующего маршрутизатора. При этом он выполняет те же действия, что и конечный узел.

Одношаговая маршрутизация обладает еще одним преимуществом – она позволяет сократить объем таблиц маршрутизации в конечных узлах и маршрутизаторах за счет использования в качестве номера сети назначения, так называемого маршрута по умолчанию – *default*, который обычно занимает в таблице маршрутизации последнюю строку. Если в таблице маршрутизации есть такая запись, то все пакеты с номерами сетей, которые отсутствуют в таблице

маршрутизации, передаются маршрутизатору, указанному в строке *default*. Поэтому маршрутизаторы часто хранят в своих таблицах ограниченную информацию о сетях интереса, пересылая пакеты для остальных сетей в порт и маршрутизатор, используемые по умолчанию. Подразумевается, что маршрутизатор, используемый по умолчанию, передаст пакет на магистральную сеть, а маршрутизаторы, подключенные к магистрали, имеют полную информацию о составе интереса.

Особенно часто приемом маршрутизации по умолчанию пользуются конечные узлы. Хотя они также в общем случае имеют в своем распоряжении таблицу маршрутизации, ее объем обычно незначителен, т. к. маршрутизация для компьютера – не основное занятие. Главная роль в маршрутизации пакетов в концепции протокола IP отводится, естественно, маршрутизаторам, которые должны обладать гораздо более полными таблицами маршрутизации, чем конечные узлы. Конечный узел часто вообще работает без таблицы маршрутизации, имея только сведения об IP-адресе маршрутизатора по умолчанию. При наличии одного маршрутизатора в локальной сети этот вариант – единственно возможный для всех конечных узлов. Но даже при наличии нескольких маршрутизаторов в локальной сети, когда проблема их выбора стоит перед конечным узлом, задание маршрута по умолчанию часто используется в компьютерах для сокращения объема их маршрутной таблицы.

Другим способом разгрузки компьютера от необходимости ведения больших таблиц маршрутизации является получение от маршрутизатора сведений о рациональном маршруте для какой-нибудь конкретной сети с помощью протокола *ICMP*.

Кроме маршрута *default*, в таблице маршрутизации могут встретиться два типа специальных записей – запись о специфичном для узла маршруте и запись об адресах сетей, непосредственно подключенных к портам маршрутизатора.

Специфичный для узла маршрут содержит вместо номера сети полный IP-адрес, т. е. адрес, имеющий ненулевую информацию не только в поле номера сети, но и в поле номера узла. Предполагается, что для такого конечного узла маршрут должен выбираться не так, как для всех остальных узлов сети, к

которой он относится. В случае, когда в таблице есть разные записи о продвижении пакетов для всей сети N и ее отдельного узла, имеющего адрес N , D , при поступлении пакета, адресованного узлу N , D , маршрутизатор отдаст предпочтение записи для N , D .

Записи в таблице маршрутизации, относящиеся к сетям, непосредственно подключенным к маршрутизатору, в поле «*Расстояние до сети назначения*» содержат нули.

Еще одним отличием работы маршрутизатора и конечного узла при выборе маршрута является способ построения таблицы маршрутизации. Если маршрутизаторы обычно автоматически создают таблицы маршрутизации, обмениваясь служебной информацией, то для конечных узлов таблицы маршрутизации создаются, как правило, вручную администраторами, и хранятся в виде постоянных файлов на дисках.

9.5. Протокол IPv6

В протоколе IPv6 (или *IPng* – *IP NEXT Generation*, *следующего поколения*) используются 128-разрядные адреса вместо 32-разрядных. Это позволяет увеличить адресное пространство в 296 раз. Ниже приведен формат основного заголовка IPv6-пакета (рис. 9.7.).

0	5	9	17	25	31
Версия	Приоритет	Метка потока			
Длина данных			Следующий заголовок	Ограничение шагов	
Адрес отправителя (128)					
Адрес получателя (128)					

Рис. 9.7. Формат заголовка дейтаграммы IPv6

Поле «*Версия*» (*Version*), 4 бита. Указывает номер используемой версии IP. Для IPv6 значение этого поля равно 6. Поле «*Приоритет*» (*Priority*), 4 бита. Используется для установки приоритета пакета. Поле «*Метка потока*» (*Traffic Class*), 24 бита. Служит для идентификации последовательности

пакетов. Все пакеты одного сообщения имеют одинаковое значение этого поля. Поле «Длина данных» (*Payload Length*), 16 бит. Показывает длину поле данных (в байтах). Поле «Следующий заголовок» (*Next Header*), 8 бит. Это поле содержит информацию о типе заголовка, который следует за заголовком IPv6. Например, 6 – для заголовка протокола TCP, 17 – для UDP и т. д. Поле «Ограничение пересылок» (*Hop Limit*), 8 бит. Имеет такое же назначение как и поле «Time to Live» в IPv4. Величина этого поля уменьшается на 1 при прохождении дейтаграммой промежуточного узла. Если величина этого поля равна 0, дейтаграмма уничтожается. Поля «Адрес отправителя» (*Source Address*) и «Адрес получателя» (*Destination Address*), длина каждого поля по 128 бит. В IPv6 128-разрядные адреса записываются в виде восьми 16-разрядных целых чисел, разделенных двоеточием.

Например, 777A:B666:0000:0000:0000:2345:ABCD:00E8.

Укороченная запись: 777A:B666:0:0:0:2345:ABCD:E8.

Выводы:

Задачу доставки сообщений между узлами составной сети решает протокол IP. Так как он является дейтаграммным, то никаких гарантий надежной доставки сообщений не дается.

Максимальная длина IP-пакета составляет 65 535 байт. Заголовок имеет длину 20 байт и содержит информацию о сетевых адресах отправителя и получателя, параметры фрагментации, время жизни пакета, контрольную сумму и некоторые другие параметры.

Вид таблицы IP-маршрутизации зависит от конкретной реализации маршрутизатора, но все они содержат два обязательных поля: поля адреса назначения и следующего маршрутизатора.

Записи в таблицу маршрутизации могут поступать из разных источников: в результате конфигурирования программное обеспечение стека TCP/IP заносит в таблицу записи о непосредственно подключенных сетях и маршрутизаторах по умолчанию, записи об особых адресах; администратор вручную заносит записи о специфических маршрутах и о маршруте по умолчанию; протоколы

маршрутизации автоматически заносят в таблицу динамические записи об имеющихся маршрутах.

Эффективное средство структуризации IP-сетей – маски, позволяющие разделить одну сеть на несколько подсетей или объединить несколько сетей в одну более крупную сеть. Технология бесклассовой междоменной маршрутизации (CIDR), решает две основные задачи: экономное расходование адресного пространства; в уменьшение числа записей в таблицах.

Важной особенностью протокола IP является его способность выполнять динамическую фрагментацию пакетов при передаче их между сетями с различными максимально допустимыми значениями длины поля данных кадров (MTU).

Контрольные вопросы:

1. Какие применяются протоколы межсетевого взаимодействия?
2. Какой формат пакета IP?
3. Какие основные функции протокола IP?
4. В чем различие между протоколами IPv4 и IPv6?
5. Какой максимальный размер поля данных IP-пакета?
6. Какова максимальная единица транспортировки сети Ethernet?
7. Какова максимальная единица транспортировки сети FDDI
8. Какова максимальная единица транспортировки сети X.25
9. Каким образом организуется управление фрагментацией?
10. Каким образом выполняется маршрутизация с помощью IP-адресов?
11. Какие способы построения таблицы маршрутизации?
12. Какие преимущества дает технология CIDR?
13. Объясните понятие метрика?
14. Какие метрики применяются для выбора оптимального маршрута?
15. Какими преимуществами обладает одношаговая маршрутизация?

10. СЕТЕВЫЕ СЛУЖБЫ

10.1. Электронная почта

Электронная почта (e-mail) – самый распространенный и самый доступный сервис Internet. Позволяет пользователям посылать письма в любую точку мира, оборудованную компьютерами, подключенными к Internet.

Работа электронной почты во многом схожа с работой обычной почты. Присутствуют также ящики для отправки почты, пункты сортировки и пересылки почты, средства доставки и, личные почтовые ящики, куда приходит почта. Корреспонденция подготавливается пользователем на своем рабочем месте: программой подготовки почты; обычным текстовым редактором. Обычно программа подготовки почты вызывает текстовый редактор, который пользователь предпочитает всем остальным программам этого типа. Затем пользователь должен вызвать программу отправки почты (программа подготовки почты вызывает программу отправки автоматически). Для пользователей персональных компьютеров, имеющих почтовые ящики на своих машинах и работающих с почтовыми серверами через коммутируемые телефонные линии, могут потребоваться дополнительные действия.

Письмо представляет собой обычный текстовый файл с заголовком в начале, состоящий из нескольких текстовых строк:

- *From: e-mail@адрес отправителя;*
- *Date: дата отправления;*
- *To: e-mail@адреса получателей;*
- *c: e-mail@адреса получателей, копии письма;*

- *Subject:* Строка с краткой аннотацией письма, а также нескольких служебных полей: маршрут, по которому шло письмо, размер, статус и т. д.

Прежде чем получить или отправить письма, пользователь должен ввести свои *реквизиты*, некоторые из которых нужны для связи с почтовым сервером, некоторые для сообщения о себе адресату. Параметры эти следующие:

- *Имя (Name)* – Имя отправителя почты в реальной жизни;
- *E-mail (Электронная почта)* – Адрес электронной почты отправителя;

- *Учетная запись* (Учетная запись *POP3, POP3 User Name*) – Часть *e-mail*- адреса до знака *@*. По этому имени почтовый сервер определяет, из какого почтового ящика он должен брать письма;

- *Пароль (Password)* – Подтверждает личность пользователя. Когда на одном компьютере работают несколько пользователей, получающих почту (например, на компьютерах учебных классов), лучше всего не вводить пароль в настройках, а оставить это поле пустым, тогда программа будет его спрашивать перед каждым получением писем, а другие пользователи не смогут воспользоваться вашим паролем и читать ваши письма.

Наибольший интерес представляет *почтовый ящик*, накапливающий приходящую почту. Почтовый ящик организуется *почтовым сервером* для хранения писем до момента, изъятия их пользователем. С понятием почтового ящика тесно связано понятие *адрес электронной почты*. Электронный адрес обозначает почтовый ящик, в который будет доставлено письмо, а, следовательно, и пользователя, который его получит. *E-mail- адрес* состоит из двух частей: *учетной записи пользователя* и *адреса почтового отделения*. Учетная запись состоит из последовательности латинских букв, цифр и некоторых знаков и уникальна для каждого пользователя, имеющего почтовый ящик на данном сервере. Адрес почтового отделения состоит из двух или более слов, разделенных точками, и идентифицирует почтовый сервер. Однако, адрес почтового отделения и имя почтового сервера – не одно и то же.

Все написанные и полученные пользователем письма почтовая программа раскладывает по *папкам*. Существуют четыре стандартные папки:

- *исходящие (outbox)* – для написанных, но еще не отправленных писем;

- *отправленные (sent)*;

- *входящие (inbox)* – для всех полученных писем;

- *удаленные (trash)* – для ненужных писем.

Кроме них пользователь может создавать и удалять свои собственные папки по своему вкусу. Письма можно перемещать из папки в папку произвольным образом. Методы перехода от

папки к папке могут существенно отличаться для каждой почтовой программы. Однако, некоторые методы присущи всем почтовым программам.

Отправка почты включает в себя:

- создание письма;
- заполнение поля адресата «Кому»;
- заполнение поля «Тема»;
- написание текста письма;
- отправка письма.

В зависимости от настроек почтовой программы письмо может быть отослано немедленно или его отправка откладывается, после чего оно помещается в папку «Отправленные» в первом случае или в папку «Исходящие» во втором. При втором варианте письма накапливаются и отправляются все разом по специальной команде пользователя.

Чтение писем инициируется по команде пользователя. Перед передачей письма на компьютер пользователя, почтовый сервер требует ввести пароль (если не задан в настройках). Полученные письма складываются в папку «Входящие». С письмом могут присылаться документы Word, графические файлы. Для их просмотра нужно дважды щелкнуть мышью по пиктограммам этих файлов. Сохранить их на диске можно, перетащив мышкой в нужный каталог или через меню.

Все составляющие почтовой службы не взаимодействует напрямую пользователем, только через специальные клиентские программы, называемые *почтовыми*. Почтовая программа запускается на рабочей станции и по требованию пользователя делает запросы серверу на чтение и отправку писем, организует работу с письмами (рис. 10.1).

В Internet используется несколько протоколов для передачи почты:

- *SMTP (Simple Mail Transfer Protocol)* отправление сообщений;
- *POP2 (Post Office Protocol Version 2)* только получение сообщений;
- *POP3 (Post Office Protocol Version 3)* отправление/получение сообщений;
- *IMAP (Interactive Mail Access Protocol)* отправление/получение сообщений.

В документации на Mail-программу указывается, с какими протоколами она работает.

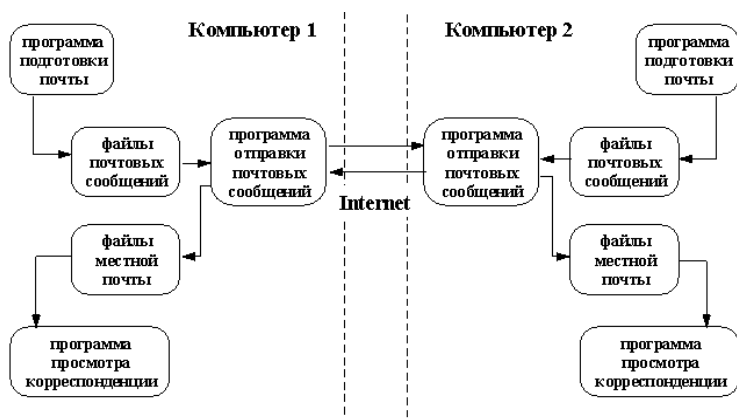


Рис. 10.1. Структура взаимодействия участников почтового обмена

10.2. Протоколы SMTP и POP3

Для работы с электронной почтой в Internet разработан специальный протокол *Simple Mail Transfer Protocol (SMTP)*, являющийся протоколом прикладного уровня и используется для обмена почтовыми сообщениями в сети Internet. *SMTP* не зависит от транспортной среды, может использоваться для доставки почты в сетях с протоколами, отличными от *TCP/IP* и *X.25*. Достигается это за счет концепции *IPCE (Inter Process Communication Environment)*. *IPCE* позволяет взаимодействовать процессам, поддерживающим *SMTP* в интерактивном режиме.

Модель протокола. Взаимодействие в рамках *SMTP* строится по принципу двусторонней связи, устанавливаемой между отправителем и получателем почтового сообщения. При этом отправитель инициирует соединение и посылает запросы на обслуживание, а получатель на эти запросы отвечает (рис. 10.2). Фактически, отправитель выступает в роли клиента, а получатель – сервера.

Канал связи устанавливается непосредственно между

отправителем и получателем сообщения, почта достигает абонента в течение нескольких секунд после отправки.

Протокол обмена почтовой информацией *POP3* (*Post Office Protocol*) предназначен для разбора почты из почтовых ящиков пользователей на их рабочие места при помощи программ-клиентов.

Если по протоколу *SMTP* пользователи отправляют корреспонденцию через Internet, то по протоколу *POP3* пользователи получают корреспонденцию из своих почтовых ящиков на почтовом сервере в локальные файлы.

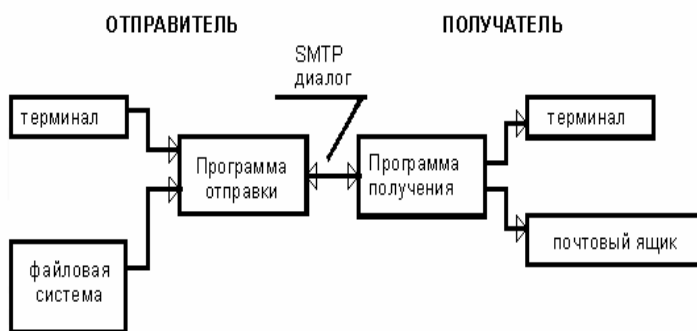


Рис. 10.2. Схема взаимодействия по протоколу SMTP

10.3. WWW. Основные компоненты технологии

Тим Бернерс Ли в 1989 г. считал, что информационная система, построенная на принципах гипертекста, должна объединить все множество информационных ресурсов CERN, состоящее из базы данных отчетов, компьютерной документации, списков почтовых адресов, информационной реферативной системы, наборов данных результатов экспериментов и т. п. Гипертекстовая технология должна была позволить легко «перепрыгивать» из одного документа в другой.

К 1989 г. гипертекст представлял новую, многообещающую

технологии, имеющую большое число реализаций с одной стороны, а с другой стороны разрабатывались формальные модели гипертекстовых систем, описательного характера, навеянные успехом реляционного подхода описания данных. Идея Т. Бернерс Ли заключалась в применении гипертекстовой модели к информационным ресурсам, распределенным в сети, сделать это максимально простым способом. Основатель трех «краеугольных камней» системы из четырех, разработал:

- язык гипертекстовой разметки документов *HTML (Hyper Text Markup Language)*;
- универсальный способ адресации ресурсов в сети *URL (Universal Resource Locator)*;
- протокол обмена гипертекстовой информацией *HTTP (Hyper Text Transfer Protocol)*;
- универсальный интерфейс шлюзов *CGI (Common Gateway Interface)*, позднее добавила команда NCSA.

Идея *HTML* – пример чрезвычайно удачного решения проблемы построения гипертекстовой системы с помощью специального средства управления отображением. На разработку языка гипертекстовой разметки существенное влияние оказали два фактора: исследования в области интерфейсов гипертекстовых систем и желание обеспечить простой и быстрый способ создания гипертекстовой базы данных, распределенной на сети.

В *HTML* гипертекстовые ссылки встроены в тело документа и хранятся как его часть. Часто в системах применяют специальные форматы хранения данных для повышения эффективности доступа. В *WWW* документы – это обычные *ASCII*-файлы, которые можно подготовить в любом текстовом редакторе.

С момента разработки первой версии языка (*HTML 1.0*) прошло много лет. За это время произошло серьезное развитие языка. Почти вдвое увеличилось число элементов разметки, развиваются средства описания не текстовых информационных ресурсов, способы взаимодействия с прикладным программным обеспечением, совершенствуется механизм разработки типовых стилей.

Универсальная **форма адресации информационных**

ресурсов Universal Resource Identification (URI) представляет собой стройную систему, учитывающую опыт адресации и идентификации *e-mail*, *Gopher*, *WAIS*, *telnet*, *ftp* и т. п. Но реально из всего, что описано в *URI*, для организации баз данных в *WWW* требуется только *Universal Resource Locator (URL)*. Без наличия этой спецификации вся мощь *HTML* оказалась бы бесполезной. *URL* используется в гипертекстовых ссылках и обеспечивает доступ к распределенным ресурсам сети. В *URL* можно адресовать другие гипертекстовые документы формата *HTML*, ресурсы *e-mail*, *telnet*, *ftp*, *Gopher*, *WAIS*. Различные интерфейсные программы по-разному осуществляют доступ к этим ресурсам. Одни, как например *Netscape*, сами способны поддерживать взаимодействие по протоколам, отличным от протокола *HTTP*, базового для *WWW*, другие, как например *Chimera*, вызывают для этой цели внешние программы. Но в первом случае, базовой формой представления отображаемой информации является *HTML*, а ссылки на другие ресурсы имеют форму *URL*. Программы обработки электронной почты в формате *MIME* также имеют возможность отображать документы, представленные в формате *HTML*, для этого зарезервирован тип «*text/html*».

Hyper Text Transfer Protocol протокол предназначен для обмена гипертекстовыми документами и учитывает специфику такого обмена. В процессе взаимодействия, клиент может получить новый адрес ресурса на сети, запросить встроенную графику, принять и передать параметры и т. п. Управление в *HTTP* реализовано в виде *ASCII*-команд. Реально разработчик гипертекстовой базы данных сталкивается с элементами протокола только при использовании внешних расчетных программ или при доступе к внешним информационным ресурсам, например базам данных.

Спецификация **Common Gateway Interface, CGI** разработана для расширения возможностей *WWW* за счет подключения всевозможного внешнего программного обеспечения. Предложенный и описанный в *CGI* способ подключения не требует дополнительных библиотек. Сервер взаимодействует с программами через стандартные потоки ввода/вывода, что упрощает программирование. При реализации

CGI чрезвычайно важное место заняли методы доступа (*GET*, *POST*), описанные в *HTTP*.

Архитектура WWW-технологии. WWW построена по схеме «клиент-сервер». Программа-клиент выполняет функции интерфейса пользователя и обеспечивает доступ практически ко всем информационным ресурсам Internet. Фактически, клиент – это интерпретатор *HTML*. И как типичный интерпретатор, клиент в зависимости от команд (разметки) выполняет различные функции. В круг этих функций входит: размещение текста на экране; обмен информацией с сервером по мере анализа полученного *HTML-текста*. При анализе *URL-спецификации* или по командам сервера клиент запускает дополнительные внешние программы для работы с документами в форматах, отличных от *HTML* (*GIF*, *JPEG*, *MPEG*, *Postscript* и т. п.).

Сервер протокола *HTTP*, базы данных документов в формате *HTML*, управляемые сервером, и программное обеспечение, разработанное в стандарте спецификации *CGI*. До образования Netscape реально использовались два *HTTP-сервера*: сервер *CERN* и сервер *NCSA*. В настоящее время число базовых серверов расширилось, появился очень сервер для *MS-Windows*, и *Apache-сервер* для *Unix-платформ* и др.

База данных HTML-документов – это часть файловой системы, содержащая текстовые файлы в формате *HTML* и связанные с ними графику и другие ресурсы. Прикладное программное обеспечение, работающее с сервером, можно разделить на программы-шлюзы и пр. *Шлюзы* – это программы, обеспечивающие взаимодействие сервера с серверами других протоколов, например *ftp*, или с распределенными на сети серверами Oracle. Прочие программы – это программы, принимающие данные от сервера и выполняющие какие-либо действия: получение текущей даты, реализацию графических ссылок, доступ к локальным базам данных или просто расчеты.

Компоненты архитектуры *WWW* существуют практически для всех типов компьютерных платформ и свободно доступны в сети. Любой, кто имеет доступ в Internet, может создать свой *WWW-сервер*, или, по крайней мере, посмотреть информацию с других серверов.

10.4. Протокол FTP

FTP архивы являются одним из основных информационных ресурсов Internet. Фактически это распределенный депозитарий текстов, программ, фильмов, фотографий, аудио записей и прочей информации, хранящейся в виде файлов на различных компьютерах по всему миру. Эта информация подразделяется на три категории:

- защищенная информация, режим доступа к ней определяется ее владельцами и разрешается по специальному соглашению с потребителем. Сюда относятся коммерческие архивы (коммерческие версии программ в архивах *fip.microsoft.com*, *fip.bsdi.com*), закрытые национальные и международные некоммерческие ресурсы (работы по международным проектам), частная некоммерческая информация со специальными режимами доступа (частные благотворительные фонды);

- информационные ресурсы ограниченного использования, программы класса *shareware* (*Trumpet Winsock*, *Atis Mail*, *Netscape* и т. п.). В данный класс могут входить ресурсы, ограниченного времени использования, ограниченного времени действия, т. е. пользователь может использовать текущую версию на свой страх и риск, но никто не будет ему оказывать поддержку (имеется пересечение со свободно распространяемыми);

- свободно распространяемые информационные ресурсы или *free ware*, если речь идет о программном обеспечении. Относится все, что можно свободно получить по сети без специальной регистрации (документация, программы и т. д.). Наиболее известными свободно распространяемыми программами являются программы проекта *GNU Free Software Foundation*. Свободно распространяемое программное обеспечение не имеет сертификата качества, но его разработчики открыты для обмена опытом.

Наиболее интересными, являются две последние категории, оформленные в виде *FTP* архивов.

Технология *FTP* была разработана в рамках проекта *ARPA* и предназначалась для обмена большими объемами информации между машинами с различной архитектурой. Главным было обеспечение надежной передачи, и поэтому с современной

точки зрения *FTP* кажется перегруженным излишними редко используемыми возможностями. «*FTP*» - *File Transfer Protocol* или просто «Протокол передачи файлов» – один из старейших в Internet и входит в стандарты. Обмен данными в *FTP* проходит по *TCP* каналу, по технологии «клиент-сервер».

В *FTP* соединение инициируется интерпретатором протокола пользователя. Управление обменом осуществляется по каналу управления в стандарте протокола *TELNET*. Команды *FTP* генерируются интерпретатором протокола пользователя и передаются на сервер. Ответы сервера отправляются пользователю по каналу управления. Пользователь имеет возможность установить контакт с интерпретатором протокола сервера и другими средствами.

Команды *FTP* определяют параметры канала передачи данных, самого процесса передачи, характер работы с удаленной и локальной файловыми системами.

При организации канала передачи данных сервер инициирует обмен данными в соответствии с согласованными в сессии управления параметрами.

Канал данных устанавливается для того же *host-a*, что и канал управления, через который ведется настройка канала данных. Канал данных может быть использован, как для приема, так и для передачи данных.

Канал управления должен быть открыт при передаче данных между машинами. В случае его закрытия передача данных прекращается.

Выводы:

Электронная почта – распределенное приложение, построенное в архитектуре клиент-сервер, главная функция – предоставление пользователям сети возможности обмена электронными сообщениями. Почтовый клиент и почтовый сервер применяют в своей работе специально разработанные для почтовых систем протоколы SMTP, POP3 и IMAP.

Важнейшей сетевой службой является World Wide Web (WWW) или Всемирная паутина; благодаря которой люди получили возможность доступа к огромному объему информации в удобном для них виде и в удобное для них время.

Клиентская часть веб-службы или браузер, представляет собой приложение, устанавливаемое на компьютере конечного пользователя, важной функцией которого является поддержание графического пользовательского интерфейса.

Веб-сервер – это программа, хранящая объекты локально в каталогах компьютера, на котором она запущена, и обеспечивающая доступ к этим объектам по URL-адресам.

Клиент и сервер веб-службы связываются через сеть по протоколу передачи гипертекста HTTP.

Файловая служба на основе протокола FTP позволяет пользователям удаленных компьютеров обмениваться файлами. Системы управления сетью позволяют в автоматическом режиме контролировать сетевой трафик и управлять коммуникационным оборудованием сети. Большинство современных систем управления сетью построены на основе протокола SNMP.

Контрольные вопросы:

1. Какие существуют сервисы Internet?
2. Что представляет собой электронная почта?
3. Какие основные компоненты у технологии World Wide Web?
4. Какие возможности предоставляет технология WWW?
5. Какие протоколы работают с технологией WWW?
6. Как взаимодействуют участники почтового обмена?
7. Что обеспечивает доступ к распределенным ресурсам сети?
8. В чем отличие между протоколами SMTP, POP3 и FTP?
9. Какие категории информации функционируют по ресурсам Internet?

11. БЕЗОПАСНОСТЬ IP-СЕТИ

11.1. Проблемы безопасности, основанные на специфике протоколов стека TCP/IP

Протоколы семейства IP являются основой построения сетей *intranet* и глобальной сети *Internet*. Несмотря на то, что разработка TCP/IP финансировалась Министерством обороны США, TCP/IP он не обладает абсолютной защищенностью и допускает различные типы *атак*.

Для того чтобы предпринять подобные атаки, злоумышленник («хакер») должен обладать контролем над одной из систем, подключенной к *Internet*. Это возможно, например, в случае, взлома какой-то системы или использования собственного компьютера, соединенного с *Internet* (для многих атак достаточно иметь *PPP-доступ*).

Рассмотрим атаки, связанные с программной реализацией. Атаки на *TCP/IP* можно разделить на два вида: пассивные; активные.

Пассивные атаки на уровне TCP. При данном типе атак злоумышленники не обнаруживают себя и не вступают напрямую во взаимодействие с другими системами. Фактически все сводится к наблюдению за доступными данными или сессиями связи. Примером может служить *подслушивание* – перехват сетевого потока и его анализ (от англ. «*sniffing*»).

Активные атаки на уровне TCP. При данном типе атак злоумышленник взаимодействует с получателем информации, отправителем и/или промежуточными системами, возможно, модифицируя и/или фильтруя содержимое *TCP/IP*-пакетов. Данные типы атак технически сложны в реализации.

Активные атаки подразделяются на две части:

- перехват и модификация сетевого потока или попыток «притвориться» другой системой;
- использование протокола *TCP/IP* для приведения системы в нерабочее состояние.

Обладая достаточными привилегиями в Unix (или используя DOS или Windows, не имеющие системы ограничений пользователей), возможно ручное (программное)

формирование *IP-пакетов* и передача их по сети. Поля заголовка пакета могут быть сформированы произвольным образом. Получив такой пакет, невозможно выяснить, откуда реально он был получен, поскольку пакеты не содержат пути их прохождения. При установке обратного адреса, не совпадающего с текущим *IP-адресом*, ответ на отосланный пакет никогда не будет получен. Возможность формирования произвольных *IP-пакетов* является ключевым пунктом для осуществления активных атак. Приведем некоторые из них:

- предсказание *TCP sequence number (IP spoofing)*. В данном случае цель притвориться другой системой, которой, «доверяет» система-жертва (в случае использования протокола *rlogin/rsh* для беспарольного входа). Метод также используется для других целей – например, для использования *SMTP* жертвы для отправки поддельных писем;

- *IP-Hijacking* – в данном случае перехватывается весь сетевой поток, модифицируется и фильтруется произвольным образом. Метод является комбинацией «подслушивания» и *IP spoofing*'а;

- пассивное сканирование – сканирование часто применяется для того, чтобы выяснить, на каких *TCP-портах* работают демоны, отвечающие на запросы из сети. Обычная программа-сканер последовательно открывает соединения с различными портами. В случае, когда соединение устанавливается, программа сбрасывает его, сообщая номер порта злоумышленнику. Данный способ легко детектируется по сообщениям демонов. Однако можно воспользоваться другим методом – пассивным сканированием (английский термин «*passive scan*»). При его использовании посылаются пакеты на все порты подряд (или по какому-то заданному алгоритму);

- «затопление» *ICMP-пакетами («ping flood»)* – программа «*ping*», предназначенная для оценки качества линии, имеет ключ для «агрессивного» тестирования. В этом режиме запросы посылаются с максимально возможной скоростью, и программа позволяет оценить, как работает сеть при максимальной нагрузке. Данная атака требует от «*крэкера*» доступа к быстрым каналам в Интернет. «*ping*» посылает *ICMP-пакет* типа *ECHO REQUEST*, выставляя в нем время и его идентификатор. Ядро

машины-получателя отвечает на подобный запрос пакетом *ICMP ECHO REPLY*. Получив его «*ping*» выдает скорость прохождения пакета. При стандартном режиме работы пакеты высылаются через некоторые промежутки времени, практически не нагружая сеть. Но в «агрессивном» режиме поток *ICMP echo request/reply-пакетов* может вызвать перегрузку небольшой линии, лишив ее способности передавать полезную информацию. Случай с «*ping*» является частным случаем более общей ситуации, связанной с перегрузкой каналов. Например, посылка множества *UDP-пакетов* на 19-й порт машины, и если она, следуя общепринятым правилам, имеет на 19-м *UDP-порту* *знакогенератор*, отвечающий на пакеты строчками по 80 байт, то возникнут большие проблемы. Может подделываться обратный адрес подобных пакетов, затрудняя его обнаружение. Еще один из вариантов атаки – посылка *ICMP echo request-пакетов* с исходным адресом, указывающем на жертву, на *broadcast-адреса* крупных сетей. В результате каждая из машин ответит на этот фальшивый запрос, и машина-отправитель получит большое количество ответов. Посылка множества *broadcast-echo requests* от имени «жертвы» на *broadcast-адреса* крупных сетей, может вызвать резкое заполнение канала «жертвы». Приметы «затопления» – резко возросшая нагрузка на сеть (или канал) и повышение количества специфических пакетов (таких, как *ICMP*);

- «локальная буря» – рассмотрим атаку на пример *UDP-бури*. По умолчанию системы поддерживают работу таких *UDP-портов*: 7 («эхо», полученный пакет отсылается назад); 19 («*знакогенератор*», в ответ на полученный пакет отправителю высылается строка знакогенератора) и других (*date etc*). Может послаться единственный *UDP-пакет*, где в качестве исходного порта будет указан 7, в качестве получателя – 19-й, а в качестве адреса получателя и отправителя будут указаны, к примеру, две машины вашей сети (или даже 127.0.0.1). Получив пакет, 19-й порт отвечает строкой, которая попадает на порт 7. Седьмой порт дублирует ее и вновь отсылает на 19, и так до бесконечности. Бесконечный цикл съедает ресурсы машин и добавляет на канал бессмысленную нагрузку. Но при первом потерянном *UDP-пакете* «буря» прекратится. Данная атака

временно выводит из строя (до перезагрузки) некоторые старые модели маршрутизаторов. Очевидно, что в бесконечный разговор могут быть вовлечены многие демоны (в случае *TCP/IP* может быть применен *TCP/IP spoofing*, в случае *UDP/ICMP* достаточно пары фальшивых пакетов);

- «затопление» *SYN-пакетами* – установка *TCP*-соединения происходит в три стадии (*3-way handshake*): клиент выбирает и передает серверу *sequence number* (*C-SYN*), в ответ на это сервер высылает клиенту пакет данных, содержащий подтверждение (*C-ACK*) и собственный *sequence number* сервера (*S-SYN*). Теперь уже клиент должен выслать подтверждение (*S-ACK*). В случае входящих соединений система отвечает на пришедший *C-SYN-пакет S-SYN/C-ACK-пакетом*, переводит сессию в состояние *SYN_RECEIVED* и заносит ее в очередь. Если в течение заданного времени от клиента не придет *S-ACK*, соединение удаляется из очереди, в противном случае соединение переводится в состояние *ESTABLISHED*. Когда очередь входных соединений уже заполнена, а система получает *SYN-пакет*, приглашающий к установке соединения, он будет проигнорирован. «Затопление» *SYN-пакетами* основано на переполнении очереди сервера, после чего сервер перестает отвечать на запросы пользователей. В различных системах работа с очередью реализована различными способами. В *BSD-системах*, каждый порт имеет свою собственную очередь размером в 16 элементов. В системах *SunOS*, такого разделения нет, и система просто располагает большой общей очередью. Для блокирования, к примеру, *WWW*-порт на *BSD* достаточно 16 *SYN-пакетов*, а для *Solaris 2.5* их количество будет гораздо больше. По истечении некоторого времени (зависит от реализации) система удаляет запросы из очереди. Но ничего не мешает послать новую порцию запросов. Таким образом, даже находясь на соединении *2400 bps*, можно посылать каждые полторы минуты по 20-30 пакетов на *FreeBSD-сервер*, поддерживая его в нерабочем состоянии (эта ошибка была скорректирована в последних версиях *FreeBSD*). Случайные обратные *IP-адреса* при формировании пакетов, затрудняют его обнаружение и фильтрацию трафика.

11.2. Межсетевой экран (firewall)

Понятия *межсетевой экран*, *брандмауэр* и *firewall* рассматриваются как синонимы.

Брандмауэр - это система или комбинация систем, позволяющая разделить сеть на две или более частей и реализовать набор правил, определяющих условия прохождения пакетов из одной части в другую (рис. 11.1). Граница проводится между локальной сетью предприятия и *INTERNET*, ее можно провести и внутри локальной сети предприятия. *Брандмауэр*, таким образом, пропускает через себя весь трафик. Для каждого проходящего пакета *брандмауэр* принимает решение пропускать его или отбросить. Для того чтобы *брандмауэр* мог принимать эти решения, ему необходимо определить набор правил.

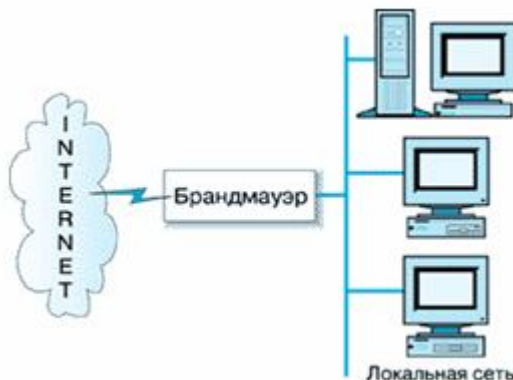


Рис. 11.1. Структура организации межсетевого экрана

Брандмауэры функционируют на какой-либо *UNIX* платформе – чаще всего это *BSDI*, *SunOS*, *AIX*, *IRIX* и т. д., реже – *DOS*, *VMS*, *WNT*, *Windows NT*. Из аппаратных платформ встречаются *INTEL*, *Sun SPARC*, *RS6000*, *Alpha*, *HP PA-RISC*, семейство *RISC* процессоров *R4400-R5000*. Помимо *Ethernet*, многие *брандмауэры* поддерживают *FDDI*, *Token Ring*. В операционную систему, под управлением которой работает

брандмауэр, вносятся изменения, цель которых – повышение защиты самого *брандмауэра*.

Эти изменения затрагивают как ядро ОС, так и соответствующие файлы конфигурации. На самом *брандмауэре* не разрешается иметь счетов пользователей (а значит и потенциальных дыр), только счет администратора. Некоторые *брандмауэры* работают только в однопользовательском режиме. Многие *брандмауэры* имеют систему проверки целостности программных кодов. При этом контрольные суммы программных кодов хранятся в защищенном месте и сравниваются при старте программы во избежание подмены программного обеспечения.

Все *брандмауэры* можно разделить на три типа:

- *пакетные фильтры (packet filter)*;
- *сервера прикладного уровня (application gateways)*;
- *сервера уровня соединения (circuit gateways)*.

Все типы могут одновременно встретиться в одном *брандмауэре*.

11.3. Пакетные фильтры

Брандмауэры с пакетными фильтрами принимают решение о том, пропускать пакет или отбросить, просматривая IP-адреса, флаги или номера *TCP* портов в заголовке этого пакета. *IP-адрес* и номер порта – это информация сетевого и транспортного уровней соответственно, но пакетные фильтры используют и информацию прикладного уровня, т. к. все стандартные сервисы в *TCP/IP* ассоциируются с определенным номером порта. Для описания правил прохождения пакетов составляются таблицы типа (рис. 11.2)

Действие	Тип пакета	Адрес источн.	Порт источн.	Адрес назнач.	Порт назнач.	Флаги
----------	------------	---------------	--------------	---------------	--------------	-------

Рис. 11.2. Шаблон таблицы описания правил

Поле «*Действие*» может принимать значения, пропускать или отбрасывать. Поле «*Тип пакета*» может содержать информацию *TCP*, *UDP* или *ICMP*. Поле «*Флаги*» – флаги из

заголовка IP-пакета. Поля «*Порт источника*» и «*Порт назначения*» имеют смысл только для *TCP* и *UDP* пакетов.

11.4. Сервера прикладного уровня и уровня соединения

Брандмауэры с серверами прикладного уровня используют сервера конкретных сервисов – *TELNET*, *FTP* и т. д. (*proxy server*), запускаемые на *брандмауэре* и пропускающие через себя весь трафик, относящийся к данному сервису. Таким образом, между клиентом и сервером образуются два соединения: от клиента до брандмауэра и от брандмауэра до места назначения.

Полный набор поддерживаемых серверов различается для каждого конкретного брандмауэра, однако чаще всего встречаются сервера для следующих сервисов:

- *терминалы (Telnet, Rlogin)*;
- *передача файлов (Ftp)*;
- *электронная почта (SMTP, POP3)*;
- *WWW (HTTP)*;
- *принтер*;
- *Rsh*;
- *Finger*;
- *новости (NNTP)* и т. д.

Использование серверов прикладного уровня позволяет решить важную задачу – скрыть от внешних пользователей структуру локальной сети, включая информацию в заголовках почтовых пакетов или службы доменных имен (*DNS*). Другим положительным качеством является возможность *аутентификации* на пользовательском уровне (*аутентификация* – процесс подтверждения идентичности чего-либо; в данном случае это процесс подтверждения, действительно ли пользователь является тем, за кого он себя выдает). При описании правил доступа используются такие параметры: название сервиса; имя пользователя; допустимый временной диапазон использования сервиса; компьютеры, с которых можно пользоваться сервисом; схемы аутентификации. Сервера протоколов прикладного уровня позволяют обеспечить наиболее высокий уровень защиты – взаимодействие с внешним миром реализуется через небольшое число прикладных

программ, полностью контролирующих весь входящий и исходящий трафик.

Сервер уровня соединения представляет собой транслятор *TCP* соединения. Пользователь образует соединение с определенным портом на брандмауэре, после чего последний производит соединение с местом назначения по другую сторону от брандмауэра. Во время сеанса этот транслятор копирует байты в обоих направлениях, действуя как провод.

Пункт назначения задается заранее, в то время как источников может быть много (*соединение типа один – много*). Используя различные порты, можно создавать различные конфигурации. Такой тип сервера позволяет создавать транслятор для любого определенного пользователем сервиса, базирующегося на *TCP*, осуществлять контроль доступа к этому сервису, сбор статистики по его использованию.

Сравнительные характеристики. К положительным качествам пакетных фильтров следует отнести следующее:

- относительно невысокая стоимость;
- гибкость в определении правил фильтрации;
- небольшая задержка при прохождении пакетов.

Недостатки следующие:

- локальная сеть видна (маршрутизируется) из *INTERNET*;
- правила фильтрации пакетов трудны в описании, требуются очень хорошие знания технологий *TCP* и *UDP*;
- при нарушении работоспособности брандмауэра все компьютеры за ним становятся полностью незащищенными либо недоступными;
- аутентификацию с использованием *IP-адреса* можно обмануть использованием *IP-спуфинга* (атакующая система выдает себя за другую, используя ее *IP-адрес*);
- отсутствует аутентификация на пользовательском уровне.

Преимуществами серверов прикладного уровня являются:

- локальная сеть невидима из *INTERNET*;
- при нарушении работоспособности брандмауэра пакеты перестают проходить через брандмауэр, тем самым не возникает угрозы для защищаемых им машин;
- защита на уровне приложений позволяет осуществлять большое количество дополнительных проверок, снижая тем

самым вероятно взлома с использованием дыр в программном обеспечении;

- аутентификация на пользовательском уровне может быть реализована системой немедленного предупреждения о попытке взлома.

Недостатками этого типа являются:

- более высокая, чем для пакетных фильтров стоимость;

- невозможность использования протоколов *Remote Procedure Call*, *RPC* и *UDP*;

- производительность ниже, чем для пакетных фильтров.

Системы сбора статистики и предупреждения об атаке. Еще одним важным компонентом брандмауэра является система сбора статистики и предупреждения об атаке. Информация обо всех событиях – отказах, входящих, выходящих соединениях, числе переданных байт, использовавшихся сервисах, времени соединения и т. д. – накапливается в файлах статистики. Многие брандмауэры позволяют гибко определять подлежащие протоколированию события, описать действия брандмауэра при атаках или попытках несанкционированного доступа – это может быть сообщение на консоль, почтовое послание администратору системы и т. д. Немедленный вывод сообщения о попытке взлома на экран консоли или администратора может помочь, если попытка оказалась успешной, и атакующий уже проник в систему. В состав многих брандмауэров входят генераторы отчетов, служащие для обработки статистики. Они позволяют собрать статистику по использованию ресурсов конкретными пользователями, по использованию сервисов, отказам, источникам, с которых проводились попытки несанкционированного доступа и т. д.

Аутентификация. Аутентификация является одним из самых важных компонентов брандмауэров. Прежде чем пользователю будет предоставлено право использовать какой-либо сервис, необходимо убедиться, что он действительно тот, за кого он себя выдает (предполагается, что этот сервис для данного пользователя разрешен: процесс определения, какие сервисы разрешены, называется *авторизацией*). Авторизация обычно рассматривается в контексте с *аутентификацией* – как только пользователь аутентифицирован, для него определяются

разрешенные ему сервисы. При получении запроса на использование сервиса от имени какого-либо пользователя, брандмауэр проверяет, какой способ аутентификации определен для данного пользователя и передает управление серверу аутентификации. После получения положительного ответа от сервера аутентификации брандмауэр образует запрашиваемое пользователем соединение. Используется принцип, получивший название «что он знает», т. е. пользователь знает некоторое секретное слово, которое он посылает серверу аутентификации в ответ на его запрос.

Одной из схем аутентификации является использование стандартных *UNIX паролей*. Эта схема является наиболее уязвимой с точки зрения безопасности – пароль может быть перехвачен и использован другим лицом.

Чаще всего используются схемы с использованием одноразовых паролей. Даже будучи перехваченным, этот пароль будет бесполезен при следующей регистрации, а получить следующий пароль из предыдущего является крайне трудной задачей. Для генерации одноразовых паролей используются как программные, так и аппаратные генераторы. *Генераторы* – устройства, вставляемые в слот компьютера. Знание секретного слова необходимо пользователю для приведения этого устройства в действие. Ряд брандмауэров поддерживают *Kerberos* – один из наиболее распространенных методов аутентификации. Некоторые схемы требуют изменения клиентского программного обеспечения – шаг, который далеко не всегда приемлем. Все коммерческие брандмауэры поддерживают несколько различных схем, позволяя администратору сделать выбор наиболее приемлемой для своих условий.

Выводы:

Информационная безопасность обеспечивается техническими средствами – системами шифрования, аутентификации, авторизации, аудита, антивирусной защиты, межсетевыми экранами и др.

Атаки на *TCP/IP* подразделяются на два вида: пассивные и активные.

При пассивной атаке на уровне TCP злоумышленники не обнаруживают себя и не вступают напрямую во взаимодействие

с другими системами, фактически все сводится к наблюдению за доступными данными или сессиями связи. Примером может служить подслушивание – перехват сетевого потока и его анализ (от англ. «sniffing»).

При активной атаке на уровне TCP злоумышленник взаимодействует с получателем информации, отправителем и/или промежуточными системами, возможно, модифицирует и/или фильтрует содержимое TCP/IP-пакетов.

Активные атаки подразделяются на две части: перехват и модификация сетевого потока или попытки «притвориться» другой системой; использование протокола TCP/IP для приведения системы в нерабочее состояние.

Аутентификация пользователя – это процедура доказательства пользователем того, что он есть тот, за кого себя выдает.

Авторизация – это процедура контроля доступа легальных пользователей к ресурсам системы и предоставление каждому из них именно тех прав, которые определены ему администратором.

Активными атаками являются: предсказание TCP sequence number, IP-Hijacking, пассивное сканирование, «затопление» ICMP-пакетами, «локальная буря», «затопление» SYN-пакетами.

Межсетевой экран, брандмауэр и firewall. Сетевой экран осуществляет информационную защиту одной части компьютерной сети от другой путем анализа проходящего между ними трафика. Сетевые экраны делятся на экраны с фильтрацией пакетов на основе IP-адресов, сетевые экраны сеансового уровня, способные фильтровать пакеты с учетом контекста, и наиболее интеллектуальные сетевые экраны прикладного уровня.

Контрольные вопросы:

1. На какие виды подразделяются атаки на TCP/IP?
2. Какие атаки относятся к пассивным?
3. Какие атаки относятся к активным?
4. Объясните виды атак: «буря», «затопление»?
5. Объясните понятие брандмауэр.
6. Какие базовые функции выполняет межсетевой экран?
7. Назначение пакетных фильтров.
8. Какие виды брандмауэров существуют, отличия?
9. В чем заключается аутентификация и авторизация?

ЗАКЛЮЧЕНИЕ

С созданием компьютерных сетей мир получил абсолютно новый способ общения. Новейшие достижения в технологии передачи данных с учетом последних изобретений в области мультимедиа позволяют обрабатывать и передавать любые массивы данных практически в любую точку земного шара. Сегодня любой обладатель портативного компьютера сможет за считанные секунды связаться со своим деловым партнером, получить необходимые данные и справку по любому интересующему его вопросу, принять участие в компьютерном совещании со своими коллегами. Компьютер стал главным средством общения между людьми.

В настоящее время большинство организаций хранит и совместно использует в сетевой среде огромные объемы жизненно важных данных. Вот почему сети сейчас так же необходимы.

Основой всемирной сети служат мощные компьютеры-серверы, которые постоянно включены и соединены друг с другом. Именно они обрабатывают и направляют поток информации, проходящей через сеть. На сегодняшний день в глобальной компьютерной сети работают миллионы таких серверов.

Сидя за монитором персонального компьютера, подключенного к сети, можно читать новости и книги, слушать музыку и радио, учиться и играть, отправлять и получать электронные письма.

Сегодня Интернет – результат успешной интеграции многих сетей в единую мировую сеть, уникальное средство, обеспечивающее оперативный доступ к необозримым кладям информации по любой тематике (на десятках тысяч информационных серверов), получения дистанционного образования и т. д.

Сети владеют всем необходимым для передачи данных: каналами связи, мощными компьютерами и специальным оборудованием, обеспечивающим адресацию информационных потоков.

Данное учебное пособие является попыткой дать четкое представление об архитектуре компьютерных сетей, принципами их построения, показать необходимость овладения знаниями, связанные с технологиями, методами организации, построения и функционирования интерсети.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Жангисина Г.Д., Смагулова Г.* Компьютерные сети. Учеб. пособие. – Алматы: АТУ, 2009. 100 с.
2. *Жангисина Г.Д.* Информатика (в 3-х частях). Учеб. пособие. –Алматы: КазАТК, 2001. 555 с.
3. *Жангисина Г.Д.* Параллельные вычисления и многопроцессорные ЭВМ в решении научно-технических задач. –Алматы: АТУ, 2006. 192с.
4. *Жангисина Г.Д.* Алгоритмизация и языки программирования. –Алматы: АТУ, 2006. 192 с.
5. *Золотов С.* Протоколы Internet. – С-Пб.: BHV-Санкт-Петербург, 1998. 304 с.
6. *Коротыгин С.* Развитие технологии беспроводных сетей: стандарт IEEE 802.11, 2001.
7. *Олифер В.Г., Олифер Н.А.* Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. – С-Пб.: Питер, 2010. 943 с.
8. *Олифер В.Г., Олифер Н.А.* Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. – С-Пб.: Питер, 2001. 672 с.
9. *Пахомов С.* Беспроводные сети семейства IEEE 802.11g. //Компьютер Пресс, 2003. № 10.
10. *Пахомов С.* Широкополосный беспроводной доступ. //Компьютер Пресс. 2004. № 5.
11. *Прохоров А.* Рынок беспроводных сетей в цифрах и фактах. //Компьютер Пресс. 2006. № 4.
12. *Столлингс В.* Современные компьютерные сети (серия «Классика computer science»). 2-изд. – С-Пб.: Питер, 2003. 783 с.
13. *Таненбаум Э.* Компьютерные сети. – С-Пб: Питер, 2003. 992 с.
14. *Ташимов М.А.* Современные вычислительные системы и сетевые технологии. –Алматы: 2004. 285 с.
15. *Тынымбаев С.* Вычислительные машины, системы, комплексы и сети. Учебник для студентов вузов. –Алма-Ата: Рауан, 1996. 290 с.

15. *Шахнович И.* Персональные беспроводные сети стандартов IEEE 802.15. 3 и 802.15.4 // *Электроника: наука, технология, бизнес.* 2004. № 6. – С. 32–36.

17. *Шульте В.* Стандарт 802.20 на мобильный широкополосный беспроводной доступ. // *LAN*, 2005. № 1.

18. Семейство стандартов IEEE 802.11. 2006. <http://www.wifi-connect.ru/>

19. IEEE 802.11, WiMAX.. с <http://ru.wikipedia.org/>.

20. IEEE 802.20. [http IEEE 802://www.softco.ru](http://www.softco.ru): ООО «Софтко».

СОДЕРЖАНИЕ

Введение.....	3
1. ЭВОЛЮЦИЯ КОМПЬЮТЕРНЫХ СЕТЕЙ.....	6
1.1. Первые глобальные сети.....	6
1.2. Первые локальные сети.....	8
1.3. Сближение локальных и глобальных сетей.....	11
1.4. Конвергенция компьютерных и телекоммуникационных сетей.....	13
Выводы.....	14
Контрольные вопросы.....	15
2. ОБЩИЕ ПРИНЦИПЫ ПОСТРОЕНИЯ СЕТЕЙ.....	16
2.1. Основные определения.....	16
2.2. Проблемы объединения нескольких компьютеров.....	18
2.2.1. Топология физических связей.....	18
2.2.2. Адресация узлов сети.....	22
2.2.3. Коммутация.....	25
Выводы.....	29
Контрольные вопросы.....	30
3. КОММУТАЦИЯ КАНАЛОВ И ПАКЕТОВ.....	32
3.1. Коммутация каналов.....	32
3.2. Коммутация пакетов.....	34
3.2.1. Принципы коммутации пакетов.....	34
3.2.2. Виртуальный канал.....	38
3.3. Коммутация сообщений.....	39
3.4. Постоянная и динамическая коммутация.....	40
Выводы.....	41
Контрольные вопросы.....	42
4. АРХИТЕКТУРА СЕТЕЙ.....	43
4.1. Декомпозиция задачи сетевого взаимодействия.....	43
4.2. Модель OSI.....	46
4.3. Стандартизация сетей.....	55
4.3.1. Стандартные стеки коммуникационных протоколов.....	55
4.3.2. Сетезависимые и сетенезависимые уровни стека	
TCP/IP.....	58
Выводы.....	59
Контрольные вопросы.....	60
5. СЕТЕВАЯ ТЕХНОЛОГИЯ ETHERNET.....	61

5.1. Методы доступа.....	62
5.1.1 Случайные методы доступа.....	62
5.1.2 Детерминированные методы доступа.....	63
5.2. Форматы кадров технологии Ethernet.....	65
5.2.1 Ethernet DIX (II).....	65
5.2.2. Стандарт IEEE 802.3.....	66
5.2.3. Fast Ethernet (IEEE 802.3u).....	67
5.2.4. Высокоскоростная технология Gigabit Ethernet.....	69
Выводы.....	71
Контрольные вопросы.....	72
6. БЕСПРОВОДНЫЕ СЕТИ.....	73
6.1. Беспроводные локальные сети.....	73
6.2. Беспроводные персональные сети.....	79
6.2.1. Стандарт IEEE 802.15.1 (Bluetooth).....	79
6.2.2. Стандарт IEEE 802.15.3.....	81
6.2.3. Стандарт IEEE 802.15.4 (ZigBee).....	83
6.3. Беспроводной широкополосный доступ BWA (Broadband Wireless Access).....	86
Выводы.....	89
Контрольные вопросы.....	90
7. ПРИНЦИПЫ ПОСТРОЕНИЯ ИНТЕРСЕТИ (INTERNETWORK).....	91
7.1. Архитектура интегрированной сети.....	91
7.2. Принципы маршрутизации.....	94
7.3. Протоколы маршрутизации.....	96
Выводы.....	100
Контрольные вопросы.....	101
8. АДРЕСАЦИЯ В СТЕКЕ ПРОТОКОЛОВ TCP/IP.....	102
8.1. Стек протоколов TCP/IP.....	102
8.2. Типы адресов стека TCP/IP.....	105
8.3. Формат записи IP-адреса.....	107
8.3.1. Классы IP-адресов.....	109
8.3.2. Особые IP-адреса.....	110
8.4. Порядок назначения IP-адресов.....	111
8.5. Отображение IP-адресов на локальные адреса.....	112
8.5.1. Протокол разрешения адресов.....	113
8.6. Система DNS.....	115
8.6.1. Символьные имена системы DNS.....	115

8.6.2. Схема работы DNS	119
8.6.3. Динамическое конфигурирование хостов.....	122
8.7. Базовые протоколы TCP/IP.....	124
Выводы.....	133
Контрольные вопросы.....	134
9. ПРОТОКОЛ МЕЖСЕТЕВОГО ВЗАИМОДЕЙСТВИЯ....	136
9.1. Формат IP-пакета	136
9.2. Пример взаимодействия узлов с использованием протокола IP.....	139
9.3. Управление фрагментацией.....	142
9.4. Маршрутизация с помощью IP-адресов.....	145
9.5. Протокол IPv6.....	150
Выводы.....	151
Контрольные вопросы.....	152
10. СЕТЕВЫЕ СЛУЖБЫ.....	153
10.1. Электронная почта	153
10.2. Протоколы SMTP и POP3.....	156
10.3. WWW. Основные компоненты технологии	157
10.4. Протокол FTP	161
Выводы.....	162
Контрольные вопросы.....	163
11. БЕЗОПАСНОСТЬ IP-СЕТИ.....	164
11.1. Проблемы безопасности, основанные на специфике протоколов стека TCP/IP.....	164
11.2. Межсетевой экран (firewall).....	168
11.3. Пакетные фильтры.....	169
11.4. Сервера прикладного уровня и уровня соединения	170
Выводы.....	173
Контрольные вопросы.....	174
Заключение.....	175
Библиографический список.....	177

Учебное издание

Жангисина Гульнур Давлетжановна
Юбузова Халича Ибрагимовна

КОМПЬЮТЕРНЫЕ СЕТИ

Учебное пособие

Нач. РО ИИЦ

Редактор

Компьютерная верстка

З. Губайдулина

Г. Дюсенбаева

Д. Тажиева

Подписано в печать 2014 г.

Тираж экз. Формат 60х84х 1/16. Бумага типогр. № 1.

Уч.-изд.л. . Усл. п.л. . Заказ № . Цена договорная.

Издание Казахского национального технического университета

им. К.И. Сатпаева

Учебно-издательский центр КазНТУ,

г. Алматы, ул. Сатпаева, 22

ISBN 978-601-228-629-8

