МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Томский государственный университет

систем управления и радиоэлектроники

Кафедра радиотехнических систем

Б.Ф. Ноздреватых

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Учебное пособие «Конспект лекций»

для студентов технических вузов

Томск, 2016

АННТОТАЦИЯ

Учебное пособие включает в себя теоретический материал для студентов технических вузов при изучении дисциплин «Информационные технологии», «Информатика» и других дисциплин, использующих сетевые информационные технологии, компьютерные сети.

Учебное пособие предназначено для подготовки студентов **направления подготовки 11.05.01 «Радиоэлектронные системы и комплексы», 11.03.02 «Инфокоммуникационные технологии и системы связи», 11.03.01 «Радиотехника» и другие.**

Содержание

Лекция 1.
Информационные технологии. Введение. Компьютерные сети.

Технология при переводе с греческого (techne) означает искусство, мастерство, умение, а это не что иное, как процессы. Под процессом следует понимать определенную совокупность действий, направленных на достижение поставленной цели. Процесс должен определяться выбранной человеком стратегией и реализоваться с помощью совокупности различных средств и методов.

Под технологией материального производства понимают процесс, определяемый совокупностью средств и методов обработки, изготовления, изменения состояния, свойств, формы сырья или материала. Технология изменяет качество или первоначальное состояние материи в целях получения материального продукта

Информация является одним из ценнейших ресурсов общества наряду с такими традиционными материальными видами ресурсов, как нефть, газ, полезные ископаемые и др., а значит, процесс ее переработки по аналогии с процессами переработки материальных ресурсов можно воспринимать как технологию. Тогда справедливо следующее определение.

Информационная технология — процесс, использующий совокупность средств и методов сбора, обработки и передачи данных (первичной информации) для получения информации нового качества о состоянии объекта, процесса или явления (информационного продукта).

Цель технологии материального производства — выпуск продукции, удовлетворяющей потребности человека или системы.

Цель информационной технологии — производство информации для ее анализа человеком и принятия на его основе решения по выполнению какого-либо действия.

Известно, что, применяя разные технологии к одному и тому же материальному ресурсу, можно получить разные изделия, продукты. То же самое будет справедливо и для технологии переработки информации.

Информационная технология является наиболее важной составляющей процесса использования информационных ресурсов общества. К настоящему времени она прошла несколько эволюционных этапов, смена которых определялась главным образом развитием научно-технического прогресса, появлением новых технических средств переработки информации. В современном обществе основным техническим средством технологии переработки информации служит персональный компьютер, который существенно повлиял как на концепцию построения и использования технологических процессов, так и на качество результатной информации. Внедрение персонального компьютера в информационную сферу и применение телекоммуникационных средств связи определили новый этап развития информационной технологии и, как следствие, изменение ее названия за счет присоединения одного из синонимов: "новая", "компьютерная" или "современная".

Прилагательное "новая" подчеркивает новаторский, а не эволюционный характер этой технологии. Ее внедрение является новаторским актом в том смысле, что она существенно изменяет содержание различных видов деятельности в организациях. В понятие новой информационной технологии включены также коммуникационные технологии, которые обеспечивают

передачу информации разными средствами, а именно — телефон, телеграф, телекоммуникации, факс и др.

Компьютерные сети отнюдь не являются единственным видом сетей, созданным человеческой цивилизацией, пример — электрические сети. В них легко можно найти аналоги компонентов любой территориальной компьютерной сети: источникам информационных ресурсов соответствуют электростанции, магистралям — высоковольтные линии электропередачи, сетям доступа — трансформаторные подстанции, клиентским терминалам — осветительные и бытовые электроприборы.

Компьютерные сети, называемые также 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 решает набор взаимосвязанных задач, обмениваясь данными в автоматическом режиме. С другой стороны, компьютерные сети могут рассматриваться как средство передачи информации на большие расстояния, для чего в них применяются методы кодирования и мультиплексирования данных, получившие развитие в различных телекоммуникационных системах.

Первые компьютеры 50-х годов большие, громоздкие и дорогие — предназначались для очень небольшого числа избранных компаний (корпораций). Часто они занимали целые здания. Такие компьютеры не были предназначены для интерактивной работы пользователя, а применялись в режиме пакетной обработки.

Системы пакетной обработки, как правило, строились на базе мэйнфрейма — мощного и надежного компьютера универсального назначения. Пользователи подготавливали перфокарты, содержащие данные и команды программ, и передавали их в вычислительный центр. Операторы вводили эти карты в компьютер, а распечатанные результаты пользователи получали обычно только на следующий день. Таким образом, одна неверно набитая карта означала как минимум суточную задержку. Конечно, для пользователей интерактивный режим работы, при котором можно с терминала оперативно руководить процессом обработки своих данных, был бы удобней. Но интересами пользователей на первых этапах развития вычислительных систем в значительной степени пренебрегали. Во главу угла ставилась эффективность работы самого дорогого устройства вычислительной машины — процессора, даже в ущерб эффективности работы использующих его специалистов.

По мере удешевления процессоров в начале 60-х годов появились новые способы организации вычислительного процесса, которые позволили учесть интересы пользователей. Начали развиваться интерактивные многотерминальные системы разделения времени. В таких системах каждый пользователь получал собственный терминал, с помощью которого он мог вести диалог с компьютером. Количество одновременно работающих с компьютером пользователей определялось его мощностью: время реакции вычислительной системы должно было быть достаточно мало, чтобы пользователю была не слишком заметна параллельная работа с компьютером других пользователей. Терминалы, выйдя за пределы вычислительного центра, рассредоточились по всему предприятию. И хотя вычислительная мощность оставалась полностью централизованной, некоторые

функции, такие как ввод и вывод данных, стали распределенными. Подобные многотерминальные централизованные системы внешне уже были очень похожи на локальные вычислительные сети. Действительно, рядовой пользователь работу за терминалом мэйнфрейма воспринимал примерно так же, как сейчас он воспринимает работу за подключенным к сети персональным компьютером. Пользователь мог получить доступ к общим файлам и периферийным устройствам, при этом у него поддерживалась полная иллюзия единоличного владения компьютером, так как он мог запустить нужную ему программу в любой момент и почти сразу же получить результат.

Однако до появления локальных сетей нужно было пройти еще большой путь, так как многотерминальные системы, хотя и имели внешние черты распределенных систем, все еще поддерживали централизованную обработку данных.

К тому же потребность предприятий в создании локальных сетей в это время еще не созрела — в одном здании просто нечего было объединять в сеть, так как из-за высокой стоимости вычислительной техники предприятия не могли себе позволить роскошь приобретения нескольких компьютеров.

А вот потребность в соединении компьютеров, находящихся на большом расстоянии друг от друга, к этому времени уже вполне назрела. Началось все с решения более простой задачи — доступа к компьютеру с терминалов, удаленных от него на многие сотни, а то и тысячи километров. Терминалы соединялись с компьютерами через телефонные сети с помощью модемов. Такие сети позволяли многочисленным пользователям получать удаленный доступ к разделяемым ресурсам нескольких мощных суперкомпьютеров. Затем

появились системы, в которых наряду с удаленными соединениями типа терминал-компьютер были реализованы и удаленные связи типа компьютер-компьютер.

Компьютеры получили возможность обмениваться данными в автоматическом режиме, что, собственно, и является базовым признаком любой вычислительной сети.

На основе подобного механизма в первых сетях были реализованы службы обмена файлами, синхронизации баз данных, электронной почты и другие ставшие теперь традиционными сетевые службы.

Итак, хронологически первыми появились глобальные сети (Wide Area Network, WAN), то есть сети, объединяющие территориально рассредоточенные компьютеры, возможно находящиеся в различных городах и странах.

Именно при построении глобальных сетей были впервые предложены и отработаны многие основные идеи, лежащие в основе современных вычислительных сетей. Такие, например, как многоуровневое построение коммуникационных протоколов, концепции коммутации и маршрутизации пакетов.

Глобальные компьютерные сети очень многое унаследовали от других, гораздо более старых и распространенных глобальных сетей — телефонных. Главное технологическое новшество, которое привнесли с собой первые глобальные компьютерные сети, состояло в отказе от принципа коммутации каналов, на протяжении многих десятков лет успешно использовавшегося в телефонных сетях.

Выделяемый на все время сеанса связи составной телефонный канал, передающий информацию с постоянной скоростью, не мог эффективно использоваться пульсирующим трафиком компьютерных данных, у

которого периоды интенсивного обмена чередуются с продолжительными паузами. Натурные эксперименты и математическое моделирование показали, что пульсирующий и в значительной степени не чувствительный к задержкам компьютерный трафик гораздо аффективней передается сетями, работающими по принципу коммутации пакетов, когда данные разделяются на небольшие порции — пакеты, — которые самостоятельно перемещаются по сети благодаря наличию адреса конечного узла в заголовке пакета. Так как прокладка высококачественных линий связи на большие расстояния обходится очень дорого, то в первых глобальных сетях часто использовались уже существующие каналы связи, изначально предназначенные совсем для других целей. Например, 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. Поскольку скорость передачи дискретных компьютерных данных по таким каналам была очень низкой (десятки килобитов 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, построенных с использованием каналов связи низкого качества, отличаются сложными процедурами контроля и восстановления данных. Типичным примером таких сетей являются сети X.25, разработанные еще в начале 70-х, когда низкоскоростные аналоговые каналы, арендуемые у телефонных компаний, были преобладающим типом каналов, соединяющих

компьютеры и коммутаторы глобальной вычислительной сети.

В 1969 году министерство обороны США инициировало работы по объединению в единую сеть суперкомпьютеров оборонных и научно-исследовательских центров. Эта сеть, получившая название ARPANET, стала отправной точкой для создания первой и самой известной ныне глобальной сети — Интернет.

Сеть ARPANET объединяла компьютеры разных типов, работавшие под управлением различных операционных систем (ОС) с дополнительными модулями, реализующими коммуникационные протоколы, общие для всех компьютеров сети. ОС этих компьютеров можно считать первыми сетевыми операционными системами.

Истинно сетевые ОС в отличие от многотерминальных ОС позволяли не только рассредоточить пользователей, но и организовать распределенные хранение и обработку данных между несколькими компьютерами, связанными электрическими связями. Любая сетевая операционная система, с одной стороны, выполняет все функции локальной операционной системы, а с другой стороны, обладает некоторыми дополнительными средствами, позволяющими ей взаимодействовать через сеть с операционными системами других компьютеров. Программные модули, реализующие сетевые функции, появлялись в операционных системах постепенно, по мер>е развития сетевых технологий, аппаратной базы компьютеров и возникновения новых задач, требующих сетевой обработки.

Прогресс глобальных компьютерных сетей во многом определялся прогрессом телефонных сетей.

С конца 60-х годов в телефонных сетях все чаше стала применяться передача голоса в цифровой форме.

Это привело к появлению высокоскоростных цифровых каналов, соединяющих автоматические телефонные станции (АТС) и позволяющих одновременно передавать десятки и сотни разговоров.

К настоящему времени глобальные сети по разнообразию и качеству предоставляемых услуг догнали локальные сети, которые долгое время лидировали в этом отношении, хотя и появились на свет значительно позже.

Важное событие, повлиявшее на эволюцию компьютерных сетей, произошло 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. Эмпирический закон Гроша перестал соответствовать действительности, так как десяток мини-компьютеров, имея ту же стоимость, что и мэйнфрейм, решали некоторые задачи (как правило, хорошо распараллеливаемые) быстрее.

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

Шло время, и потребности пользователей вычислительной техники росли. Их уже не удовлетворяла изолированная работа на собственном компьютере, им хотелось в автоматическом режиме обмениваться

компьютерными данными с пользователями других подразделений. Ответом на эту потребность стало появление первых локальных вычислительных сетей (рис. 1.5).

Локальные сети (Local Area Network, LAN) — это объединения компьютеров, сосредоточенных на небольшой территории, обычно в радиусе не более 1-2 км, хотя в отдельных случаях локальная сеть может иметь и большие 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), а также 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 т. п., могли соединять только те конкретные модели компьютеров, для которых были разработаны, например, мини- компьютеры PDP-11 с мэйнфреймом IBM 360 или мини-компьютеры HP с микрокомпьютерами LSI-11. Такая ситуация создала большой простор для творчества студентов — названия многих курсовых и дипломных проектов начинались тогда со слов «Устройство сопряжения...».

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.

Мощным стимулом для их появления послужили персональные компьютеры. Эти массовые продукты стали идеальными элементами построения сетей — с одной стороны, они были достаточно мощными, чтобы обеспечивать работу сетевого программного обеспечения, а с другой — явно нуждались в объединении своей вычислительной мощности для решения сложных задач, а также разделения дорогих периферийных устройств и дисковых массивов. Поэтому персональные компьютеры стали преобладать в локальных сетях, причем не только в качестве клиентских компьютеров, но и в качестве центров хранения и обработки данных, то есть сетевых серверов, потеснив с этих привычных ролей мини- компьютеры и мэйнфреймы.

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 превратили процесс построения локальной сети из решения нетривиальной технической проблемы в рутинную работу. Для создания сети достаточно было приобрести стандартный кабель, сетевые адаптеры соответствующего стандарта, например Ethernet, вставить адаптеры в компьютеры, присоединить их к кабелю стандартными разъемами и установить на компьютеры одну из популярных сетевых операционных систем, например Novell NetWare.

Разработчики локальных сетей привнесли много нового в организацию работы пользователей. Так, стало намного проще и удобнее, чем в глобальных сетях,

получать доступ к общим сетевым ресурсам. Последствием и одновременно движущей силой такого прогресса стало появление огромного числа непрофессиональных пользователей, освобожденных от необходимости изучать специальные (и достаточно сложные) команды для сетевой работы.

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

Простые алгоритмы работы предопределяют низкую стоимость оборудования Ethernet. Широкий диапазон иерархии скоростей позволяет рационально строить локальную сеть, выбирая ту технологию семейства, которая в наибольшей степени отвечает 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.

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 (10, 16 и 100 Мбит/с) в то время была существенно выше, чем в глобальных (от 2,4 Кбит/с до 2 Мбит/с).
- Разнообразие услуг. Высокие скорости обмена данными позволили предоставлять в локальных сетях широкий спектр услуг — это, прежде всего, разнообразные механизмы использования файлов, хранящихся на дисках других компьютеров сети, совместное использование устройств печати, модемов, факсов, доступ к единой базе данных, электронная почта и др. В то же время глобальные сети в основном ограничивались почтовыми и файловыми услугами в их простейшем (не самом удобном для пользователя) виде.

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

Сближение в методах передачи данных происходит на платформе цифровой передачи данных по волоконно-оптическим линиям связи. Эта среда передачи используется практически во всех технологиях локальных сетей для скоростного обмена информацией на расстояниях свыше 100 метров, на ней же построены современные магистрали первичных сетей SDH и DWDM, предоставляющих свои цифровые каналы для объединения оборудования глобальных компьютерных сетей.

Высокое качество цифровых каналов изменило требования к протоколам глобальных компьютерных сетей. На первый план вместо процедур обеспечения надежности вышли процедуры обеспечения

гарантированной средней скорости доставки информации пользователям, а также механизмы приоритетной обработки пакетов особенно чувствительного к задержкам трафика, например голосового. Эти изменения нашли отражение в новых технологиях глобальных сетей, таких как Frame Relay и ATM. В этих сетях предполагается, что искажение битов происходит настолько редко, что ошибочный пакет выгоднее просто уничтожить, а все проблемы, связанные с его потерей, перепоручить программному обеспечению более высокого уровня, которое непосредственно не входит в состав сетей Frame Relay и ATM. Большой вклад в сближение локальных и глобальных сетей внесло доминирование протокола IP. Этот протокол сегодня работает поверх любых технологий локальных и глобальных сетей (Ethernet, Token Ring, ATM, Frame Relay), объединяя различные подсети в единую составную сеть.

Начиная с 90-х годов компьютерные глобальные сети, работающие на основе скоростных цифровых каналов, существенно расширили спектр предоставляемых услуг и догнали в этом отношении локальные сети. Стало возможным создание служб, работа которых связана с доставкой пользователю больших объемов информации в реальном времени — изображений, видеофильмов, голоса, в общем, всего того, что получило название мультимедийной информации. Наиболее яркий пример — гипертекстовая информационная служба World Wide Web, ставшая основным поставщиком информации в Интернете. Ее интерактивные возможности превзойти возможности многих аналогичных служб ло-кальных сетей, так что разработчикам локальных сетей пришлось просто позаимствовать эту службу у глобальных сетей. Процесс переноса технологий из глобальной сети Интернет в локальные приобрел такой массовый характер, что

появился даже специальный термин — intranet-технологии (intra — внутренний).

В локальных сетях в последнее время уделяется такое же большое внимание методам обеспечения защиты информации от несанкционированного доступа, как и в глобальных. Это обусловлено тем, что локальные сети перестали быть изолированными, чаше всего они имеют выход в «большой мир» через глобальные связи.

И наконец, появляются новые технологии, изначально предназначенные для обоих видов сетей. Ярким представителем нового поколения технологий является технология ATM, которая может служить основой как глобальных, так и локальных сетей, эффективно объединяя все существующие типы трафика в одной транспортной сети. Другим примером является 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.

Еще одним признаком сближения локальных и глобальных сетей является появление сетей, занимающих промежуточное положение между локальными и глобальными сетями.

Городские сети, или сети мегаполисов (Metropolitan Area Network, MAN), предназначены для обслуживания территории крупного города.

Эти сети используют цифровые линии связи, часто оптоволоконные, со скоростями на магистрали от 155 Мбит/с и выше. Они обеспечивают экономичное соединение локаль-ных сетей между собой, а также выход в глобальные сети. Сети MAN первоначально были разработаны только для передачи данных, но сейчас перечень предоставляемых ими услуг расширился, в

частности они поддерживают видеоконференции и интегральную передачу голоса и текста. Современные сети MAN отличаются разнообразием предоставляемых услуг, позволяя своим клиентам объединять коммуникационное оборудование различного типа, в том числе офисные АТС.

Конвергенция компьютерных и телекоммуникационных сетей

С каждым годом усиливается тенденция сближения компьютерных и телекоммуникационных сетей разных видов. Предпринимаются попытки создания универсальной, так называемой мультисервисной сети, способной предоставлять услуги как компьютерных, так и телекоммуникационных сетей.

К телекоммуникационным сетям относятся телефонные сети, радиосети и телевизионные сети. Главное, что объединяет их с компьютерными сетями, — то, что в качестве ресурса, предоставляемого клиентам, выступает информация. Однако имеется некоторая специфика, касающаяся вида, в котором представляют информацию компьютерные и телекоммуникационные сети. Так, изначально компьютерные сети разрабатывались для передачи алфавитно-цифровой информации, которую часто называют просто данными, поэтому у компьютерных сетей имеется и другое название — сети передачи данных, в то время как телекоммуникационные сети были созданы для передачи только голосовой информации (и изображения в случае телевизионных сетей).

Сегодня мы являемся свидетелями конвергенции телекоммуникационных и компьютерных сетей, которая идет по нескольким направлениям.

Прежде всего, наблюдается сближение видов услуг, предоставляемых клиентам. Первая и не очень успешная попытка создание мультисервисной сети, способной оказывать различные услуги, в том числе услуги

телефонии и передачи данных, привела к появлению технологии цифровых сетей с интегрированным обслуживанием (Integrated Services Digital Network, ISDN). Однако на практике ISDN предоставляет сегодня в основном телефонные услуги, а на роль глобальной мультисервисной сети нового поколения, часто называемой в ан¬глоязычной литературе Next Generation Network (NGN), или New Public Network (NPN), претендует Интернет. Интернет будущего должен обладать возможностью оказывать все виды телекоммуникационных услуг, в том числе новые виды комбинированных услуг, в которых сочетаются несколько традиционных услуг, например услуга универсальной службы сообщений, объединяющей электронную почту, телефонию, факсимильную службу и пейджинговую связь. Наибольших успехов на практическом поприще достигла 1Р-телефония, услугами которой прямо и ли косвенно сегодня пользуются миллионы людей. Однако для того чтобы стать сетью NGN, Интернету еще предстоит пройти большой путь.

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

Телефония уже давно сделала ряд шагов навстречу компьютерным сетям, прежде всего, за счет представления голоса в цифровой форме, что делает принципиально возможным передачу телефонного и компьютерного трафика по одним и тем же цифровым каналам (телевидение также может сегодня передавать изображение в цифровой форме). Телефонные сети широко используют комбинацию методов коммутации каналов и пакетов. Так, для передачи служебных сообщений (называемых сообщениями сигнализации)

применяются протоколы коммутации пакетов, аналогичные протоколам компьютерных сетей, а для передачи собственно голоса между абонентами коммутируется традиционный составной канал.

Дополнительные услуги телефонных сетей, такие как переадресация вызова, конференц- связь, телеголосование и другие, могут создаваться с помощью так называемой интеллектуальной сети (Intelligent Network, IN), по своей сути являющейся компьютерной сетью с серверами, на которых программируется логика услуг.

Сегодня пакетные методы коммутации постепенно теснят традиционные для телефонных сетей методы коммутации каналов даже при передаче голоса. У этой тенденции есть достаточно очевидная причина — на основе метода коммутации пакетов можно более эффективно использовать пропускную способность каналов связи и коммутационного оборудования. Например, паузы в телефонном разговоре могут составлять до 40 % общего времени соединения, однако только пакетная коммутация позволяет «вырезать» паузы и использовать высвободившуюся пропускную способность канала для передачи трафика других абонентов. Другой веской причиной перехода к коммутации пакетов является популярность Интернета — сети, построенной на основе данной технологии.

Обращение к технологии коммутации пакетов для одновременной передачи через пакетные сети разнородного трафика — голоса, видео и текста — сделало актуальным разработку новых методов обеспечения требуемого качества обслуживания (Quality of Service, QoS). Методы QoS призваны минимизировать уровень задержек для чувствительного к ним трафика, например голосового, и одновременно гарантировать среднюю

скорость и динамичную передачу пульсаций для трафика данных.

Однако неверно было бы говорить, что методы коммутации каналов морально устарели и у них нет будущего. На новом витке спирали развития они находят свое применение, но уже в новых технологиях.

Компьютерные сети тоже многое позаимствовали у телефонных и телевизионных сетей. В частности, они берут на вооружение методы обеспечения отказоустойчивости телефон-ных сетей, за счет которых последние демонстрируют высокую степень надежности, так недостающую порой Интернету и корпоративным сетям.

Сегодня становится все более очевидным, что мультисервисная сеть нового поколения не может быть создана в результате «победы» какой-нибудь одной технологии или одного подхода. Ее может породить только процесс конвергенции, когда от каждой технологии будет взято все самое лучшее и соединено в некоторый новый сплав, который и даст требуемое качество для поддержки существующих и создания новых услуг. Появился новый термин — инфокоммуникационная сеть, который прямо говорит о двух составляющих современной сети — информационной (компьютерной) и телекоммуникационной. Учитывая, что новый термин еще не приобрел достаточной популярности, мы будем использовать устоявшийся термин «телекоммуникационная сеть» в расширенном значении, то есть включать в него и компьютерные сети.

Лекция 2. Общие принципы построения сетей

Исторически главной целью объединения компьютеров в сеть было разделение ресурсов: пользователи компьютеров, подключенных к сети, или приложения, выполняемые на этих компьютерах, получают возможность автоматического доступа к разнообразным ресурсам остальных компьютеров сети, к числу которых относятся:

- периферийные устройства, такие как диски, принтеры, плоттеры, сканеры и др.;
- данные, хранящиеся в оперативной памяти или на внешних запоминающих устройствах;
- вычислительная мощность (за счет удаленного запуска «своих» программ на «чужих» компьютерах).

Чтобы обеспечить пользователей разных компьютеров возможностью совместного использования ресурсов сети, компьютеры необходимо оснастить некими дополнительными сетевыми средствами.

Для связи устройств в них, прежде всего, должны быть предусмотрены внешние интерфейсы.

Интерфейс — в широком смысле — формально определенная логическая и/или физическая граница между взаимодействующими независимыми объектами Интерфейс задает параметры, процедуры и характеристики взаимодействия объектов.

Разделяют физический и логический интерфейсы.

- Физический интерфейс (называемый также портом) — определяется набором электрических связей и характеристиками сигналов. Обычно он представляет собой разъем с набором контактов, каждый из которых имеет определенное назначение, например, это может быть группа

контактов для передачи данных, контакт синхронизации данных и т. п. Пара разъемов соединяется кабелем, состоящим из набора проводов, каждый из которых соединяет соответствующие контакты. В таких случаях говорят о создании линии, или канала, связи между двумя устройствами.

- Логический интерфейс (называемый также протоколом) — это набор информационных сообщений определенного формата, которыми обмениваются два устройства или две программы, а также набор правил, определяющих логику обмена этими сообщениями.

При объединении в сеть большего числа компьютеров возникает целый комплекс новых проблем.

Топология физических связей

Объединяя в сеть несколько (больше двух) компьютеров, необходимо решить, каким образом соединить их друг с другом, другими словами, выбрать конфигурацию физических связей, или топологию.

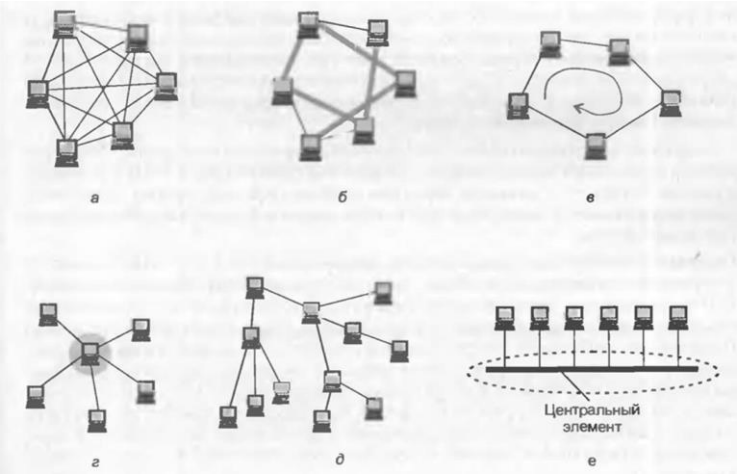
Число возможных вариантов конфигурации резко возрастает при увеличении числа связываемых устройств. Мы можем соединять каждый компьютер с каждым или же связывать их последовательно, предполагая, что они будут общаться, передавая сообщения друг другу «транзитом». Транзитные узлы должны быть оснащены специальными средствами, позволяющими им выполнять эту специфическую посредническую операцию. В качестве транзитного узла может выступать как универсальный компьютер, так и специализированное устройство.

От выбора топологии связей существенно зависят характеристики сета. Например, наличие между узлами нескольких путей повышает надежность сети и делает

возможным распределение загрузки между отдельными каналами. 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 неполно связные.

Полносвязная топология соответствует сети, в которой каждый компьютер непосредственно связан со всеми остальными. Несмотря на логическую простоту, этот вариант оказывается громоздким и неэффективным. Действительно, в таком случае каждый компьютер в сети должен иметь большое количество коммуникационных портов, достаточное для связи с каждым из остальных компьютеров сети. Для каждой пары компьютеров должна быть выделена отдельная физическая линия связи. (В некоторых случаях даже две, если невозможно использование этой линии для двусторонней передачи.) Полносвязные топологии в крупных сетях применяются редко, так как для связи N узлов требуется N(N- 1)/2 физических дуплексных линий связей, то есть имеет место квадратичная зависимость от числа узлов. Чаще этот вид топологии используется в многомашинных комплексах или в 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.

Ячеистая топология получается из полносвязной путем удаления некоторых связей. Ячеистая топология допускает соединение большого количества компьютеров и характерна, как правило, для крупных сетей.

В сетях с кольцевой топологией данные перелаются по кольцу от одного компьютера к другому. Главным достоинством кольца является то, что оно по своей природе обеспечивает резервирование связей. Действительно, любая пара узлов соединена здесь двумя путями — по часовой стрелке и против нее. Кроме того, кольцо представляет собой очень удобную конфигурацию для организации обратной связи — данные, сделав полный оборот, возвращаются к узлу-источнику. Поэтому источник может контролировать процесс доставки данных адресату. Часто это свойство кольца используется для тестирования связности сет и поиска узла, работающего некорректно. В то же время в сетях с кольцевой топологией необходимо принимать специальные меры,

чтобы в случае выхода из строя или отключения какого-либо компьютера не прерывался канал связи между остальными узлами кольца. Звездообразная топология образуется в случае, когда каждый компьютер подключается непосредственно к общему центральному устройству, называемому концентратором . В функции концентратора входит направление передаваемой компьютером информации одному или всем остальным компьютерам сети. В качестве концентратора может выступать как универсальный компьютер, так и специализированное устройство. К недостаткам звездообразной топологии относится более высокая стоимость сетевого оборудования из-за необходимости приобретения специализированного центрального устройства. Кроме того, возможности по наращиванию количества узлов в сети ограничиваются количеством портов концентратора.

Иногда имеет смысл строить сеть с использованием нескольких концентраторов, иерархически соединенных между собой звездообразными связями. Получаемую в результате структуру называют иерархической звездой, или деревом. В настоящее время дерево является самой распространенной топологией связей как в локальных, так и глобальных сетях.

Особым частным случаем звезды является общая шина. Здесь в качестве центрального элемента выступает пассивный кабель, к которому по схеме «монтажного ИЛИ» подключается несколько компьютеров (такую же топологию имеют многие сети, использующие беспроводную связь — роль общей шины здесь играет общая радиосреда). Передаваемая информация распространяется по кабелю и доступна одновременно всем компьютерам, присоединенным к этому кабелю. Основными преимуществами такой схемы являются ее

дешевизна и простота присоединения новых узлов к сети, а недостатками - низкая надежность (любой дефект кабеля полностью парализует всю сеть) и невысокая производительность (в каждый момент времени только один компьютер может передавать данные по сети, поэтому пропускная способность делится здесь между всеми узлами сети).

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

Еще одной новой проблемой, которую нужно учитывать при объединении трех и более компьютеров, является проблема их адресации, точнее адресации их сетевых интерфейсов . 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 следующим образом:

- 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;
- адрес произвольной рассылки (anycast), определенный в новой версии протокола IPv6, 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 и символьными (site.domen.ru).

Символьные адреса (имена) предназначены для запоминания людьми и поэтому обычно несут смысловую нагрузку. Для работы в больших сетях символьное имя может иметь иерархическую структуру, например ftp-arch1 .ucl.ac.uk. Этот адрес говорит о том, что данный компьютер поддерживает ftp-архив в сети одного из колледжей Лондонского университета (University College London — ucl) и эта сеть относится к академической ветви (ac) Интернета Великобритании (United Kingdom — uk). При работе в пределах сети Лондонского университета такое длинное символьное имя явно избыточно и вместо него можно пользоваться кратким символьным именем ftp-arch 1. 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 обычно используется только аппаратурой, поэтому его стараются сделать по возможности компактным и записывают в виде двоичного или

шестнадцатеричного числа. При задании MAC-адресов не требуется выполнение ручной работы, так как они обычно встраиваются в аппаратуру компанией-изготовителем, поэтому их называют также аппаратными адресами (hardware address). Использование плоских адресов является жестким решением — при замене аппаратуры, например сетевого адаптера, изменяется и адрес сетевого интерфейса компьютера.

При иерархической организации адресное пространство структурируется в виде вложенных друг в друга подгрупп, которые, последовательно сужая адресуемую область, в конце концов, определяют отдельный сетевой интерфейс.

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 требуется уже 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 обычно применяют сразу несколько схем адресации, так что сетевой интерфейс компьютера может одновременно иметь несколько адресов-имен. Каждый адрес задействуется в той ситуации, когда соответствующий вид адресации наиболее удобен. А для преобразования адресов из одного вида в другой используются специальные вспомогательные протоколы, которые называют протоколами разрешения адресов. Пользователи адресуют компьютеры иерархическими символьными именами, которые автоматически заменяются в сообщениях, передаваемых по сети, иерархическими числовыми

адресами. С помощью этих числовых адресов сообщения доставляются из одной сети в другую, а после доставки сообщения в сеть назначения вместо иерархического числового адреса используется плоский аппаратный адрес компьютера. Проблема установления соответствия между адресами различных типов может решаться как централизованными, так и распределенными средствами.

При централизованном подходе в сети выделяется один или несколько компьютеров (серверов имен), в которых хранится таблица соответствия имен различных типов, например символьных имен и числовых адресов. Все остальные компьютеры обращаются к серверу имен с запросами, чтобы по символьному имени найти числовой номер необходимого компьютера.

При распределенном подходе каждый компьютер сам хранит все назначенные ему адреса разного типа. Тогда компьютер, которому необходимо определить по известному иерархическому числовому адресу некоторого компьютера его плоский аппаратный адрес, посылает в сеть широковещательный запрос. Все компьютеры сети сравнивают содержащийся в запросе адрес с собственным. Тот компьютер, у которого обнаружилось совпадение, посылает ответ, содержащий искомый аппаратный адрес. Такая схема использована в протоколе разрешения адресов (Address Resolution Protocol, ARP) стека TCP/IP Достоинство распределенного подхода состоит в том, что он позволяет отказаться от выделения специального компьютера в качестве сервера имен, который, к тому же, часто требует ручного задания таблицы соответствия адресов. Недостатком его является необходимость широковещательных сообщений, перегружающих сеть. Именно поэтому распределенный подход используется в небольших сетях, а централизованный — в больших. До сих нор мы говорили об адресах сетевых интерфейсов,

компьютеров и коммуникационных устройств, однако 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. Очевидно, что достаточно обеспечить уникальность адреса процесса в пределах компьютера. Примером адресов процессов являются номера портов TCP и UDP, используемые в стеке TCP/IP

Итак, пусть компьютеры физически связаны между собой в соответствии с некоторой топологией и выбрана система адресации. Остается нерешенной самая важная проблема: каким способом передавать данные между конечными узлами? Особую сложность приобретает эта задача для неполносвязной топологии сети, когда обмен данными между произвольной парой конечных узлов (пользователей) должен идти в общем случае через транзитные узлы.

Соединение конечных узлов через сеть транзитных узлов называют коммутацией. Последовательность узлов, лежащих на пути от отправителя к получателю, образует маршрут.

В самом общем виде задача коммутации может быть представлена в виде следующих взаимосвязанных частных задач.

1. Определение информационных потоков, для которых требуется прокладывать маршруты.

2. Маршрутизация потоков.

3. Продвижение потоков, то есть распознавание потоков и их локальная коммутация на каждом транзитном узле.

4.Мультиплексирование и демультиплексирование потоков.

Определение информационных потоков

Информационным потоком, или потоком данных, называют непрерывную последовательность данных, объединенных набором общих признаков, выделяющих эти данные из общего сетевого трафика.

Например, как поток можно определить все данные, поступающие от одного компьютера; объединяющим признаком в данном случае служит адрес источника. Эти же данные можно представить как совокупность нескольких подпотоков, каждый из которых в качестве дифференцирующего признака имеет адрес назначения. Наконец, каждый из этих подпотоков, в свою очередь, можно разделить на более мелкие подпотоки, порожденные разными сетевыми приложениями — электронной почтой, профаммой копирования файлов, веб-сервером. Данные, образующие поток, могут быть представлены в виде различных информационных единиц данных — пакетов, кадров или ячеек.

Очевидно, что при коммутации в качестве обязательного признака выступает адрес назначения данных. На основании этого признака весь ноток входящих в транзитный узел данных разделяется на подпотоки, каждый из которых передается на интерфейс, соответствующий маршруту продвижения данных.

Адреса источника и назначения определяют поток для пары соответствующих конечных узлов. Однако часто бывает полезно представить этот поток в виде нескольких подпотоков, причем для каждого из них может быть проложен свой особый маршрут. Рассмотрим пример, когда на одной и той же паре конечных узлов выполняется несколько взаимодействующих по сети приложений, каждое из которых предъявляет к сети свои особые

требования. В таком случае выбор маршрута должен осуществляться с учетом характера передаваемых данных, например, для файлового сервера важно, чтобы передаваемые им большие объёмы данных направлялись по каналам, обладающим высокой пропускной способностью, а для программной системы управления, которая посылает в сеть короткие сообщения, требующие обязательной и немедленной отработки, при выборе маршрута более важна надежность линии связи и минимальный уровень задержек на маршруте. Кроме того, даже для данных, предъявляющих к сети одинаковые требования, может прокладываться несколько маршрутов, чтобы за счет распараллеливания ускорить передачу данных.

Метка потока — это особый тип признака. Она представляет собой некоторое число, которое несут все данные потока. Глобальная метка назначается данным потока и не меняет своего значения на всем протяжении его пути следования от узла источника до узла назначения, таким образом, она уникально определяет ноток в пределах сети. В некоторых технологиях используются локальные метки потока, динамически меняющие свое значение при передаче данных от одного узла к другому.

Таким образом, распознавание потоков во время коммутации происходит на основании признаков, в качестве которых, помимо обязательного адреса назначения данных, могут выступать и другие признаки, такие, например, как идентификаторы приложений.

Задача маршрутизации, в свою очередь, включает в себя две подзадачи:

- определение маршрута;
- оповещение сети о выбранном маршруте.

Определить маршрут означает выбрать последовательность транзитных узлов и их интерфейсов,

через которые надо передавать данные, чтобы доставить их адресату. Определение маршрута — сложная задача, особенно когда конфигурация сети такова, что между нарой взаимодействующих сетевых интерфейсов существует множество путей. Чаще всего выбор останавливают на одном оптимальном но некоторому критерию маршруте. В качестве критериев оптимальности могут выступать, например, номинальная пропускная способность и загруженность каналов связи; задержки, вносимые каналами; количество промежуточных транзитных узлов; надежность каналов и транзитных узлов.

Но даже в том случае, когда между конечными узлами существует только один путь, при сложной топологии сети его нахождение может представлять собой нетривиальную задачу.

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. Однако эмпирический подход к определению маршрутов мало пригоден для большой сети со сложной топологией. В этом случае используются автоматические методы определения маршрутов. Для этого 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.

Абстрактный способ измерения степени близости между двумя объектами называется метрикой. Так, для измерения длины маршрута могут быть использованы разные метрики - количество транзитных узлов, как в предыдущем примере, линейная протяженность маршрута и даже его стоимость в денежном выражении. Для построения метрики, учитывающей пропускную способность, часто используют следующий прием: длину каждого канала-участка характеризуют величиной, обратной его пропускной способности. Чтобы оперировать целыми числами, выбирают некоторую константу, заведомо большую, чем пропускные способности каналов в сети.

Описанные подходы к выбору маршрутов не учитывают текущую степень загруженности каналов трафиком. Используя аналогию с автомобильным трафиком, можно сказать, что мы выбирали маршрут по карте, учитывая количество промежуточных городов и ширину дороги (аналог пропускной способности канала), отдавая предпочтение скоростным магистралям.

После того как маршрут определен (вручную или автоматически), надо оповестить о нем все устройства сети. Сообщение о маршруте должно нести каждому транзитному устройству примерно такую информацию: «каждый раз, когда в устройство поступят данные, относящиеся к потоку п, их следует передать для дальнейшего продвижения на интерфейс. Каждое подобное сообщение о маршруте обрабатывается устройством, в результате создается новая запись в таблице коммутации. В этой таблице локальному или глобальному признаку (признакам) потока (например, метке, номеру входного интерфейса или адресу

назначения) ставится в соответствие номер интерфейса, на который устройство должно передавать данные, относящиеся к этому потоку.

Конечно, детальное описание структуры сообщения о маршруте и содержимого таблицы коммутации зависит от конкретной технологии, однако эти особенности не меняют сущности рассматриваемых процессов.

Передача информации транзитным устройствам о выбранных маршрутах, так же как и определение маршрута, может осуществляться вручную или автоматически. Администратор сети может зафиксировать маршрут, выполнив в ручном режиме конфигурирование устройства, например, жестко скоммутировав на длительное время определенные нары входных и выходных интерфейсов (как работали «телефонные барышни» на первых коммутаторах). Он может также по собственной инициативе внести запись о маршруте в таблицу ком мутации.

Однако поскольку топология и состав информационных потоков могут меняться (отказы узлов или появление новых промежуточных узлов, изменение адресов или определение новых потоков), гибкое решение задач определения и задания маршрутов предполагает постоянный анализ состояния сети и обновление маршрутов и таблиц коммутации. В таких случаях задачи прокладки маршрутов, как правило, не могут быть решены без достаточно сложных программных и аппаратных средств.

Итак, пусть маршруты определены, записи о них сделаны в таблицах всех транзитных узлов, все готово к выполнению основной операции — передаче данных между абонентами (коммутации абонентов).

Для каждой пары абонентов эта операция может быть представлена несколькими (по числу транзитных

узлов) локальными операциями коммутации. 11режде всего, отправитель должен выставить данные на тот свой интерфейс, с которого начинается найденный марш-рут, а все транзитные узлы должны соответствующим образом выполнить «переброску» данных с одного своего интерфейса на другой, другими словами, выполнить коммутацию интерфейсов. Устройство, функциональным назначением которого является коммутация, называется коммутатором.

Однако прежде чем выполнить коммутацию, коммутатор должен распознать ноток. Для этого поступившие данные анализируются на предмет наличия в них признаков какого- либо из потоков, заданных в таблице коммутации. Если произошло совпадение, то эти данные направляются на интерфейс, определенный для них в маршруте.

Термины «коммутация», «таблица коммутации» и «коммутатор» в телекоммуникационных сетях могут трактоваться неоднозначно. Мы уже определили коммутацию как процесс соединения абонентов сети через транзитные узлы. Этим же термином мы обозначаем и соединение интерфейсов в пределах отдельного транзитного узла. Коммутатором в широком смысле называется устройство любого тина, способное выполнять операции переключения потока данных с одного интерфейса на другой. Операция коммутации может выполняться в соответствии с различными правилами и алго-ритмами. Некоторые способы коммутации и соответствующие им таблицы и устройства получили специальные названия. Например, в технологиях сетевого уровня, таких как IP и IPX, для обозначения аналогичных понятий используются термины «маршрутизация», «таблица маршрутизации», «маршрутизатор». В то же время за другими специальными типами коммутации и

соответствующими устройствами закрепились те же самые названия «коммутация», «таблица коммутации» и «коммутатор», применяемые в узком смысле, например, как коммутация и коммутатор локальной сети. Для телефонных сетей, которые появились намного раньше компьютерных, также характерна аналогичная терминология, коммутатор является здесь синонимом телефонной станции. Из-за солидного возраста и гораздо большей (пока) распространенности телефонных сетей чаще всего в телекоммуникациях под термином «коммутатор» понимают именно телефонный коммутатор.

Коммутатором может быть как специализированное устройство, так и универсальный компьютер со встроенным программным механизмом коммутации, в этом случае коммутатор называется программным. Компьютер может совмещать функции коммутации данных с выполнением своих обычных функций как конечного узла. Однако во многих случаях более рациональным является решение, в соответствии с которым некоторые узлы в сети выделяются специально для коммутации. Эти узлы образуют коммутационную сеть, к которой подключаются все остальные.

Чтобы определить, на какой интерфейс следует передать поступившие данные, коммутатор должен выяснить, к какому потоку они относятся. Эта задача должна решаться независимо от того, поступает на вход коммутатора только один «чистый» поток или «смешанный» поток, являющийся результатом агрегирования нескольких потоков. В последнем случае к задаче распознавания потоком добавляется задача демультиплексирования.

Демультиплексирование — разделение суммарного агрегированного потока на несколько отдельных потоков.

Как правило, операцию коммутации сопровождает также обратная операция мультиплексирования.

Мультиплексирование — образование из нескольких отдельных потоков общего агрегированного потока, который передается по одному физическому каналу связи.

Другими словами, мультиплексирование — это способ разделения одного имеющегося физического канала между несколькими одновременно протекающими сеансами связи между абонентами сети.

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

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

Технология мультиплексирования должна позволять получателю такого суммарного потока выполнять обратную операцию — разделение (демультиплексирование) данных на слагаемые потоки. Вообще говоря, на каждом интерфейсе могут одновременно выполняться обе функции — мультиплексирование и демультиплексирование.

Частный случай коммутатора, у которого все входящие информационные потоки коммутируются на

один выходной интерфейс, где они мультиплексируются в один агрегированный поток, называется мультиплексором. Коммутатор, который имеет один входной интерфейс и несколько выходных, называется демультиплексором.

Во всех рассмотренных ранее примерах мультиплексирования потоков к каждой линии связи подключались только два интерфейса. В том случае, когда линия связи является дуплексным каналом связи, каждый из интерфейсов монопольно использует канал связи в направлении «от себя». Это объясняется тем, что дуплексный канал состоит из двух независимых сред передачи данных (подканалов), и так как только передатчик интерфейса является активным устройством, а приемник пассивно ожидает поступления сигналов от приемника, то конкуренции подканалов не возникает. Такой режим использования среды передачи данных является в настоящее время основным в компьютерных локальных и глобальных сетях.

Однако если в глобальных сетях такой режим использовался всегда, то в локальных сетях до середины 90-х годов преобладал другой режим, основанный на разделяемой среде передачи данных.

В наиболее простом случае эффект разделения среды возникает при соединении двух интерфейсов с помощью полудуплексного канала связи, то есть такого канала, который может передавать данных в любом направлении, но только попеременно. В этом случае к одной и той же среде передачи данных (например, к коаксиальному кабелю или общей радиосреде) подключены два приемника двух независимых узлов сети.

Разделяемой средой (shared medium) называется физическая среда передачи данных, к которой непосредственно подключено несколько передатчиков узлов сети. Причем в каждый момент времени только один

из передатчиков какого-либо узла сети получает доступ к разделяемой среде и использует ее для передачи данных приемник* другого узла подключенному к этой же среде.

При таком применении среды передачи данных возникает новая задача совместного использования среды независимыми передатчиками таким образом, чтобы в каждый отдельный момент времени по среде передавались данные только одного передатчика. Другими словами, возникает необходимость в механизме синхронизации доступа интерфейсов к разделяемой среде.

Существуют различные способы решения задачи организации совместного доступа к разделяемым линиям связи. Одни из них подразумевают централизованный подход, когда доступом к каналу управляет специальное устройство — арбитр, другие — децентрализованный. Если мы обратимся к организации работы компьютера, то увидим, что доступ к системной шине компьютера, которую совместно используют внутренние блоки компьютера, управляется централизованно — либо процессором, либо специальным арбитром шины.

В сетях организация совместного доступа к сетям связи имеет свою специфику из-за существенно большего времени распространения сигналов по линиям связи. Здесь процедуры согласования доступа к линии связи могут занимать слишком большой промежуток времени и приводить к значительным потерям производительности сети. Именно по этой причине механизм разделения среды в глобальных сетях практически не используется. На первый взгляд может показаться, что механизм разделения среды очень похож на механизм мультиплексирования потоков — в том и другом случаях по линии связи передаются несколько потоков данных. Однако здесь есть принципиальное различие, касающееся того, как контролируется (управляется) линия связи. При

мультиплексировании дуплексная линия связи в каждом направлении находится под полным контролем одного коммутатора, который решает, какие потоки разделяют общий канал связи.

Для локальных сетей разделяемая среда сравнительно долго была основным механизмом использования каналов связи, который применялся во всех технологиях локальных сетей — Ethernet, ArcNet, Token Ring, FDDI. При этом в технологиях локальных сетей применялись децентрализованные методы доступа к среде, не требующие наличия арбитра в сети. Популярность техники разделения среды в локальных сетях объяснялась простотой и экономичностью аппаратных решений. Например, для создания сети Ethernet на коаксиальном кабеле никакого другого сетевого оборудования кроме сетевых адаптеров компьютеров и самого кабеля не требуется. Наращивание количества компьютеров в локальной сети Ethernet на коаксиальном кабеле выполняется также достаточно просто — путем присоединения нового отрезка кабеля к существующему.

Сегодня в проводных локальных сетях метод разделения среды практически перестал применяться. Основной причиной отказа от разделяемой среды явилась ее низкая и плохо предсказуемая производительность, а также плохая масштабируемость. Низкая производительность объясняется тем, что пропускная способность канала связи делится между всеми компьютерами сети. Например, если локальная сеть Ethernet состоит из 100 компьютеров, а для их связи используются коаксиальный кабель и сетевые адаптеры, работающие на скорости 10 Мбит/с, то в среднем на каждый компьютер приходится только 0,1 Мбит/с пропускной способности. Более точно оценить долю пропускной способности, приходящуюся на какой-либо

компьютер сети, трудно, так как эта величина зависит от многих случайных факторов, например активности других компьютеров. Наверно, к этому моменту читателю уже понятна причина плохой масштабируемости подобной сети — чем больше мы добавляем компьютеров, тем меньшая доля пропускной способности достается каждому компьютеру сети.

Описанные недостатки являются следствием самого принципа разделения среды, поэтому преодолеть их полностью невозможно. Появление в начале 90-х недорогих коммутаторов локальных сетей привело к настоящей революции в этой области, и постепенно коммутаторы вытеснили разделяемую среду полностью.

Сегодня механизм разделения среды используется только в беспроводных локальных сетях, где среда — радиоэфир — естественным образом соединяет все конечные узлы, находящиеся в зоне распространения сигнала.

Комплекс технических решений обобщенной задачи коммутации в своей совокупности составляет основу любой сетевой технологии. Как уже отмечалось, к этим 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.

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

основополагающих, к которым относят коммутацию каналов и коммутацию пакетов.

Каждый из этих двух подходов имеет свои достоинства и недостатки. Существуют традиционные области применения каждой из техник коммутации, например, телефонные сети строились и продолжают строиться с использованием техники коммутации каналов, а компьютерные сети в подавляющем большинстве основаны на технике коммутации пакетов. Техника коммутации пакетов гораздо моложе своей конкурентки и пытается вытеснить ее из некоторых областей, например ид телефонии (в форме интернет- или IP-телефонии), но этот спор пока не решен, и, скорее всего, две техники коммутации будут сосуществовать еще долгое время, дополняя друг фуга. Тем не менее по долгосрочным прогнозам многих специалистов будущее принадлежит технике коммутации пакетов, как более гибкой и универсальной.

Лекция 3. Коммутация каналов и коммутация пакетов

Исторически коммутация каналов появилась намного раньше коммутации пакетов и ведет свое происхождение от первых телефонных сетей. Невозможность динамического перераспределения пропускной способности физического канала является принципиальным ограничением сети с коммутацией каналов.

Принцип коммутации пакетов был изобретен разработчиками компьютерных сетей. При коммутации пакетов учитываются особенности компьютерного трафика, поэтому данный способ коммутации является более эффективным для компьютерных сетей по сравнению с традиционным методом коммутации каналов, применяющимся в телефонных сетях.

Однако достоинства и недостатки любой сетевой технологии — относительны. Наличие буферной памяти в коммутаторах пакетных сетей позволяет эффективно использовать пропускную способность каналов при передаче пульсирующего трафика, но приводит к случайным задержкам в доставке пакетов, что является недостатком для трафика реального времени, который традиционно передается с помощью техники коммутации каналов.

Существуют три метода продвижения пакетов, используемые в сетях с коммутацией пакетов: дейтаграммная передача, передача с установлением логического соединения и техника виртуальных каналов.

Сети, построенные на принципе коммутации каналов, имеют богатую историю, они и сегодня нашли широкое применение в мире телекоммуникаций, являясь основой создания высокоскоростных магистральных каналов связи. Первые сеансы связи между компьютерами были осуществлены через телефонную сеть, то есть также

с применением техники коммутации каналов, а пользователи, которые получают доступ в Интернет по модему, продолжают обслуживаться этими сетями, так как их данные доходят до оборудования провайдера местной телефонной сети.

В сетях с коммутацией каналов решаются все те частные задачи коммутации, которые были сформулированы ранее. Так, в качестве информационных потоков в сетях с коммутацией каналов выступают данные, которыми обмениваются пары абонентов.

Соответственно глобальным признаком потока является пара адресов (телефонных номеров) абонентов, связывающихся между собой. Для всех возможных потоков заранее определяются маршруты. Маршруты в сетях с коммутацией каналов задаются либо «вручную» администратором сети, либо находятся автоматически с привлечением специальных программных и аппаратных средств. Маршруты фиксируются н таблицах, в которых признакам потока ставятся в соответствие идентификаторы выходных интерфейсов коммутаторов. На основании этих таблиц происходит продвижение и мультиплексирование данных. Однако, как уже было сказано, в сетях с коммутацией каналов решение всех этих задач имеет свои особенности.

Одной из особенностей сетей с коммутацией каналов является понятие элементарного канала.

Элементарный канал (или просто канал) — это базовая техническая характеристика сети с коммутацией каналов, представляющая собой некоторое фиксированное в пределах данного типа сетей значение пропускной способности. Любая лини і связи в сети с коммутацией каналов имеет пропускную способность, кратную элементарному каналу, принятому для этого типа сети.

Говоря о сетях с коммутацией каналов, мы придаем термину «канал» значение единицы пропускной способности.

Значение элементарного канала, или, другими словами, минимальная единица пропускной способности линии связи, выбирается с учетом разных факторов. Очевидно, однако, что элементарный канал не стоит выбирать меньше минимально необходимой пропускной способности для передачи ожидаемой предложенной нагрузки. Например, в традиционных телефонных сетях наиболее распространенным значением элементарного канала сегодня является скорость 64 Кбит/с — это минимально достаточная скорость для качественной цифровой передачи голоса.

Задача оцифровывания голоса является частным случаем более общей проблемы — передачи аналоговой информации в дискретной форме. Она была решена в 60-е годы, когда голос начал передаваться по телефонным сетям в виде последовательности единиц и нулей. Такое преобразование основано на дискретизации непрерывных процессов как по амплитуде, так и по времени.

Амплитуда исходной непрерывной функции измеряется заданным периодом — за счет этого происходит дискретизация по времени. Затем каждый замер представляется в виде двоичного числа определенной разрядности, что означает дискретизацию по значениям — непрерывное множество возможных значений амплитуды заменяется дискретным множеством ее значений. Для качественной передачи голоса используется частота квантования амплитуды звуковых колебаний в 8000 Гц (дискретизация по времени с интервалом 125 мкс). Для представления амплитуды одного замера чаще всего используется 8 бит кода, что дает 256 градаций звукового сигнала (дискретизация по

значениям). В этом случае для передачи одного голосового канала необходима пропускная способность 64 Кбит/с: 8000 х 8 = 64 000 бит/с или 64 Кбит/с. Такой голосовой канал называют элементарным каналом цифровых телефонных сетей.

Так, линии связи, подключающие абонентов к телефонной сети, могут содержать 2, 24 или 30 элементарных каналов, а линии, соединяющие коммутаторы, — 480 или 1920 каналов.

Связь, построенную путем коммутации (соединения) элементарных каналов называют составным каналом.

Подчеркнем следующие свойства составного канала

- составной канал на всем своем протяжении состоит из одинаковою количества элементарных каналов;
- составной канал имеет постоянную и фиксированную пропускную способность на всем своем протяжении;
- составной канал создается временно на период сеанса связи двух абонентов;
- на время сеанса связи все элементарные каналы, входящие в составной канал, поступают в исключительное пользование абонентов, для которых был создан этот составной канал;
- в течение всего сеанса связи абоненты могут посылать в сеть данные со скоростью, не превышающей пропускную способность составного канала;
- данные, поступившие в составной канал, гарантированно доставляются вызываемому абоненту без задержек, потерь и с той же скоростью (скоростью источника) вне зависимости

от того, существуют ли в это время в сети и другие соединения или нет;

- после окончания сеанса связи элементарные каналы, входившие в соответствующий составной канал, объявляются свободными и возвращаются в пул распределяемых ресурсов для использования другими абонентами.

В сети может одновременно происходить несколько сеансов связи (обычная ситуация для телефонной сети, в которой одновременно передаются разговоры сотен и тысяч абонентов). Разделение сети между сеансами связи происходит на уровне элементарных каналов.

Мультиплексирование означает, что абоненты вынуждены конкурировать за ресурсы, в данном случае за элементарные каналы. Возможны ситуации, когда некоторая промежуточная линия связи уже исчерпала свободные элементарные каналы, тогда новый сеанс связи, маршрут которого пролегает через данную линию связи, не может состояться.

Для того чтобы распознать такие ситуации, обмен данными в сети с коммутацией каналов предваряется процедурой установления соединения. В соответствии с этой процедурой абонент, являющийся инициатором сеанса связи, посылает в коммутационную сеть запрос, представляющий собой сообщение, в котором содержится адрес вызываемого абонента.

Цель запроса — проверить, можно ли образовать составной канал между вызывающим и вызываемым абонентами. А для этого требуется соблюдение двух условий: наличие требуемого числа свободных элементарных каналов в каждой линии связи, и незанятость вызываемого абонента в другом соединении. Запрос перемещается по маршруту, определенному для информационного потока данной пары абонентов. При

этом используются глобальные таблицы коммутации, ставящие в соответствие глобальному признаку потока (адресу вызываемого абонента) идентификатор выходного интерфейса коммутатора (как уже упоминалось, такие таблицы часто называют также таблицами маршрутизации).

Если в результате прохождения запроса от абонента А к абоненту В выяснилось, что ничто не препятствует установлению соединения, происходит фиксация составного канала. Для этого во всех коммутаторах вдоль пути от А до В создаются записи в локальных таблицах коммутации, в которых указывается соответствие между локальными признаками потока — номерами элементарных каналов, зарезервированных для этого сеанса связи. Только после этого составной канал считается установленным, и абоненты А и В могут начать свой сеанс связи.

Таким образом, продвижение данных в сетях с коммутацией каналов происходит в два этапа:

1. В сеть поступает служебное сообщение — запрос, который несет адрес вызываемого абонента и организует создание составного канала.
2. По подготовленному составному каналу передается основной поток данных, для передачи которого уже не требуется никакой вспомогательной информации, в том числе адреса вызываемого абонента. Коммутация данных в коммутаторах выполняется на основе локальных признаков — номеров элементарных каналов.

Запросы на установление соединения не всегда завершаются успешно. Если на пути между вызывающим и вызываемым абонентами отсутствуют свободные элементарные каналы или вызываемый узел занят, то происходит отказ в установлении соединения. При отказе в установлении соединения сеть информирует вызывающего

абонента специальным сообщением. Чем больше нагрузка на сеть, то есть чем больше соединений она в данный момент поддерживает, тем больше вероятность отказа в удовлетворении запроса на установление нового соединения. Мы описали процедуру установления соединения в автоматическом динамическом режиме, основанном на способности абонентов отправлять в сеть служебные сообщения — запросы на установление соединения и способности узлов сети обрабатывать такие сообщения. Подобный режим используется телефонными сетями: телефонный аппарат генерирует запрос, посылая в сеть импульсы (или тоновые сигналы), кодирующие номер вызываемого абонента, а сеть либо устанавливает соединение, либо сообщает об отказе сигналами «занято».

Однако это — не единственно возможный режим работы сети с коммутацией каналов, существует и другой статический ручной режим установления соединения. Этот режим характерен для случаев, когда необходимо установить составной канал не на время одного сеанса связи абонентов, а на более долгий срок. Создание такого долговременного канала не могут инициировать абоненты, он создается администратором сети. Очевидно, что статический ручной режим малопригоден для традиционной телефонной сети с ее короткими сеансами связи, однако он вполне оправдан для создания высокоскоростных телекоммуникационных каналов между городами и странами на более-менее постоянной основе.

Технология коммутации каналов ориентирована на минимизацию случайных событий в сети, то есть это технология, стремящаяся к детерминизму. Во избежание всяких возможных неопределенностей значительная часть работы по организации информационного обмена выполняется заранее, еще до того, как начнется собственно передача данных. Сначала по заданному адресу

проверяется доступность необходимых элементарных каналов на всем пути от отправителя до адресата. Затем эти каналы закрепляются на все время сеанса для исключительного использования двумя абонентами и коммутируются в один непрерывный «трубопровод» (составной канал), имеющий «шлюзовые задвижки» на стороне каждого из абонентов. После этой исчерпывающей подготовительной работы остается сделать самое простое: «открыть шлюзы» и позволить информационному потоку свободно и без помех «перетекать» между заданными точками сети.

Сети с коммутацией каналов наиболее эффективно передают пользовательский трафик в том случае, когда скорость его постоянна в течение всего сеанса связи и максимально соответствует фиксированной пропускной способности физических линий связи сети. Эффективность работы сети снижается, когда информационные потоки, генерируемые абонентами, приобретают пульсирующий характер.

Так, разговаривая по телефону, люди постоянно меняют темп речи, перемежая быстрые высказывания паузами. В результате соответствующие «голосовые» информационные потоки становятся неравномерными, а значит, снижается эффективность передачи данных. Правда, в случае телефонных разговоров это снижение оказывается вполне приемлемым и позволяет широко использовать сети с коммутацией каналов для передачи голосового трафика.

Гораздо сильнее снижает эффективность сети с коммутацией каналов передача так называемого компьютерного трафика, то есть трафика, генерируемого приложениями, с которыми работает пользователь компьютера. Этот трафик практически всегда является пульсирующим. Например, когда вы загружаете из

Интернета очередную страницу, скорость трафика резко возрастает, а после окончания загрузки падает практически до нуля. Если для описанного сеанса доступа в Интернет вы задействуете сеть с коммутацией каналов, то большую часть времени составной канал между вашим компьютером и веб-сервером будет простаивать. В то же время часть производительности сети окажется закрепленной за вами и останется недоступной другим пользователям сети. Сеть в такие периоды похожа на пустой эскалатор метро, который движется, но полезную работу не выполняет, другими словами, «перевозит воздух».

Для эффективной передачи неравномерного компьютерного трафика была специально разработана **техника коммутации пакетов.**

Сети с коммутацией пакетов, так же как и сети с коммутацией каналов, состоят из коммутаторов, связанных физическими линиями связи. Однако передача данных в этих сетях происходит совершенно по-другому. Образно говоря, по сравнению с сетью с коммутацией каналов сеть с коммутацией пакетов ведет себя менее «ответственно». Например, она может принять данные для передачи, не заботясь о резервировании линий связи на пути следования этих данных и не гарантируя требуемую пропускную способность. Сеть с коммутацией пакетов не создает заранее для своих абонентов отдельных, выделенных исключительно для них каналов связи. Данные могут задерживаться и даже теряться по пути следования.

Важнейшим принципом функционирования сетей с коммутацией пакетов является представление информации, передаваемой по сети, в виде структурно отделенных друг от друга порций данных, называемых пакетами.

Каждый пакет снабжен заголовком, в котором содержится адрес назначения и другая вспомогательная информация (длина поля данных, контрольная сумма и др.), используемая для доставки пакета адресату. Наличие адреса в каждом пакете является одним из важнейших особенностей техники коммутации пакетов, так как каждый пакет может быть обработан коммутатором независимо от других пакетов, составляющих сетевой трафик. Помимо заголовка у пакета может иметься еще одно дополнительное поле, размещаемое в конце пакета и поэтому называемое концевиком. В концевике обычно помещается контрольная сумма, которая позволяет проверить, была ли искажена информация при передаче через сеть или нет.

В зависимости от конкретной реализации технологии коммутации пакетов пакеты могут иметь фиксированную или переменную длину, кроме того, может меняться состав информации, размещенной в заголовках пакетов. Например, в технологии ATM пакеты (называемые там ячейками) имеют фиксированную длину, а в технологии Ethernet установлены лишь минимально и максимально возможные размеры пакетов (кадров). Пакеты поступают в сеть без предварительного резервирования линий связи и не с фиксированной заранее заданной скоростью, как это делается в сетях с коммутацией каналов, а в том темпе, в котором их генерирует источник. Предполагается, что сеть с коммутацией пакетов, в отличие от сети с коммутацией каналов, всегда готова принять пакет от конечного узла.

Как и в сетях с коммутацией каналов, в сетях с коммутацией пакетов для каждого из потоков вручную или автоматически определяется маршрут, фиксируемый в хранящихся на коммутаторах таблицах коммутации. Пакеты, попадая на коммутатор, обрабатываются и

направляются по тому или иному маршруту на основании информации, содержащейся в их заголовках, а также в таблице коммутации.

ПРИМЕЧАНИЕ: Процедура резервирования пропускной способности может применяться и в пакетных сетях. Однако основная идея такого резервирования принципиально отличается от идеи резервирования пропускной способности в сетях с коммутацией каналов. Разница заключается в том, что пропускная способность канала сети с коммутацией пакетов может динамически перераспределяться между информационными потоками в зависимости от текущих потребностей каждого потока, чего не может обеспечить техника коммутации каналов.

Пакеты, принадлежащие как одному и тому же, так и разным информационным потокам, при перемещении по сети могут «перемешиваться» между собой, образовывать очереди и «тормозить» друг друга. На пути пакетов могут встретиться линии связи, имеющие разную пропускную способность. В зависимости от времени суток может сильно меняться и степень загруженности линий связи. В таких условиях не исключены ситуации, когда пакеты, принадлежащими одному и тому же потоку, могут перемещаться по сети с разными скоростями и даже прийти к месту назначения не н том порядке, в котором они были отправлены.

Разделение данных на пакеты позволяет передавать неравномерный компьютерный трафик более эффективно, чем в сетях с коммутацией каналов. Это объясняется тем, что пульсации трафика от отдельных компьютеров носят случайный характер и распределяются во времени так, что их пики чаще всего не совпадают. Поэтому когда линия связи передает трафик большого количества конечных узлов, то в суммарном потоке пульсации сглаживаются, и

пропускная способность линии используется более рационально, без длительных простоев.

Неопределенность и асинхронность перемещения данных в сетях с коммутацией пакетов предъявляет особые требования к работе коммутаторов в таких сетях.

Главное отличие пакетных коммутаторов от коммутаторов в сетях с коммутацией каналов состоит в том, что они имеют внутреннюю буферную память для временного хранения пакетов.

Действительно, пакетный коммутатор не может принять решения о продвижении пакета, не имея в своей памяти всего пакета. Коммутатор проверяет контрольную сумму, и только если она говорит о том, что данные пакета не искажены, начинает обрабатывать пакет и по адресу назначения определяет следующий коммутатор. Поэтому каждый пакет последовательно бит за битом помещается во входной буфер. Имея в виду это свойство, говорят, что сети с коммутацией пакетов используют технику сохранения с продвижением (store-and-forward). Заметим, что для этой цели достаточно иметь буфер размером в один пакет. Коммутатору нужны буферы для согласования скоростей передачи данных в линиях связи, подключенных к его интерфейсам. Действительно, если скорость поступления пакетов из одной линии связи в течение некоторого периода превышает пропускную способность той линии связи, в которую эти пакеты должны быть направлены, то во избежание потерь пакетов на целевом интерфейсе необходимо организовать выходную очередь. Буферизация необходима пакетному коммутатору также для согласования скорости поступления пакетов со скоростью их коммутации. Если коммутирующий блок не успевает обрабатывать пакеты (анализировать заголовки и перебрасывать пакеты на нужный интерфейс), то на интерфейсах коммутатора возникают входные очереди.

Очевидно, что для храпения входной очереди объем буфера должен превышать размер одного пакета. Существуют различные подходы к построению коммутирующего блока. Традиционный способ основан на одном центральном процессоре, который обслуживает все входные очереди коммутатора. Такой способ построения может приводить к большим очередям, так как производительность процессора разделяется между несколькими очередями. Современные способы построения коммутирующего блока основаны на многопроцессорном подходе, когда каждый интерфейс имеет свой встроенный процессор для обработки пакетов. Кроме того, существует центральный процессор, координирующий работу интерфейсных процессоров. Использование интерфейсных процессоров повышает производительность коммутатора и уменьшает очереди во входных интерфейсах. Однако такие очереди все равно могут возникать, так как центральный процессор по-прежнему остается «узким местом». Поскольку объем буферов в коммутаторах ограничен, иногда происходит потеря пакетов из-за переполнения буферов при временной перегрузке части сети, когда совпадают периоды пульсации нескольких информационных потоков Для сетей с коммутацией пакетов потеря пакетов является обычным явлением, и для компенсации таких потерь в данной сетевой технологии предусмотрен ряд специальных механизмов, которые мы рассмотрим позже.

Пакетный коммутатор может работать на основании одного из трех методов продвижения пакетов:

дейтаграммная передача;

передача с установлением логического соединения;

передача с установлением виртуального канала.

Дейтаграммный способ передачи данных основан на том, что все передаваемые пакеты продвигаются (передаются от одного узла сети другому) независимо друг от друга на основании одних и тех же правил.

Процедура обработки пакета определяется только значениями параметров, которые он несет в себе, и текущим состоянием сети (например, в зависимости от ее нагрузки пакет может стоять в очереди на обслуживание большее или меньшее время). Однако никакая информация об уже переданных пакетах сетью не хранится и в ходе обработки очередного пакета во внимание не принимается. То есть каждый отдельный пакет рассматривается сетью как совершенно независимая единица передачи — дейтаграмма.

Решение о продвижении пакета принимается на основе таблицы коммутации, ставящей в соответствие адресам назначения пакетов информацию, однозначно определяющую следующий по маршруту транзитный (или конечный) узел. В качестве такой информации могут выступать идентификаторы интерфейсов данного коммутатора или адреса входных интерфейсов коммутаторов, следующих по маршруту.

В таблице коммутации для одного и того же адреса назначения может содержаться несколько записей, указывающих соответственно на различные адреса следующего коммутатора. Такой подход называется балансом нагрузки и используется для повышения производительности и надежности сети. Некоторая «размытость» путей следования пакетов с одним и тем же адресом назначения через сеть является прямым следствием принципа независимой обработки каждого пакета, присущего дейтаграммному методу. Пакеты, следующие по одному и тому же адресу назначения, могут добираться до него разными путями также вследствие

изменения состояния сети, например отказа промежуточных коммутаторов.

Дейтаграммный метод работает быстро, так как никаких предварительных действий перед отправкой данных проводить не требуется. Однако при таком методе трудно проверить факт доставки пакета узлу назначения. Этот метол не гарантирует доставку пакета, он делает это по мере возможности — для описания такого свойства используется термин доставка с максимальными усилиями (best effort).

Следующий рассматриваемый нами способ продвижения пакетов основывается на знании устройствами сети «истории» обмена данными, например, на запоминании узлом- отправителем числа отправленных, а узлом-получателем — числа полученных пакетов. Такого рода информация фиксируется в рамках логического соединения.

Процедуре согласования двумя конечными узлами сети некоторых параметров процесса обмена пакетами называется установлением логического соединения. Параметры, о которых договариваются два взаимодействующих узла, называются параметрами логического соединения.

Наличие логического соединения позволяет более рационально по сравнению с дейтаграммным способом обрабатывать пакеты. Например, при потере нескольких предыдущих пакетов может быть снижена скорость отправки последующих. Или благодаря нумерации пакетов и отслеживанию номеров отправленных и принятых пакетов можно повысить надежность путем отбрасывания дубликатов, упорядочивания поступивших и повторения передачи потерянных пакетов.

Параметры соединения могут быть: постоянными, то есть не изменяющимися в течение всего соединения

(например, идентификатор соединения, способ шифрования пакета или максимальный размер поля данных пакета), или переменными, то есть динамически отражающими текущее состояние соединения (например, последовательные номера передаваемых пакетов).

Когда отправитель и получатель фиксируют начало нового соединения, они, прежде всего, «договариваются» о начальных значениях параметров процедуры обмена и только после этого начинают передачу собственно данных.

Передача с установлением соединения более надежна, но требует больше времени для передачи данных и вычислительных затрат от конечных узлов.

Процедура установления соединения состоит обычно из трех шагов.

1. Узел-инициатор соединения отправляет узлу-получателю служебный пакет с предложением установить соединение.

2. Если узел-получатель согласен с этим, то он посылает в ответ другой служебный пакет, подтверждающий установление соединения и предлагающий некоторые параметры, которые будут использоваться в рамках данного логического соединения. Это могут быть, например, идентификатор соединения, количество кадров, которые можно отправить без получения подтверждения и т. п.

3. Узел-инициатор соединения может закончить процесс установления соединения отправкой третьего служебного пакета, в котором сообщит, что предложенные параметры ему подходят.

Логическое соединение может быть рассчитано на передачу данных как в одном направлении — от инициатора соединения, так и в обоих направлениях. После передачи некоторого законченного набора данных, например определенного файла, узел-отправитель

инициирует разрыв данного логического соединения, посылая соответствующий служебный кадр.

Заметим, что, в отличие от передачи дейтаграммного типа, в которой поддерживается только один тип кадра — информационный, передач; с установлением соединения должна поддерживать как минимум два типа кадров — деформационные кадры переносят собственно пользовательские данные, а служебные предназначаются для установления (разрыва) соединения.

После того как соединение установлено и все параметры согласованы, конечные узлы начинают передачу собственно данных. Пакеты данных обрабатываются коммутаторами точно так же, как и при дейтаграммной передаче: из заголовков пакетов извлекаются адреса назначения и сравниваются с записями в таблицах коммутации, содержащих информацию о следующих шагах по маршруту. Так же как дейтаграммы, пакеты, относящиеся к одному логическому соединению, в некоторых случаях (например, при отказе линии связи) могут доставляться адресату по разным маршрутам.

Однако передача с установлением соединения имеет важное отличие от дейтаграммной передачи, поскольку в ней помимо обработки пакетов на коммутаторах имеет место дополнительная обработка пакетов на конечных узлах. Например, если при установлении соединения была оговорена передача данных в зашифрованном виде, то шифрование пакетов выполняется узлом-отправителем, а расшифровка — узлом-получателем. Аналогично, для обеспечения в рамках логического соединения надежности всю работу по нумерации пакетов, отслеживанию номеров доставленных и недоставленных пакетов, посылки копий и отбрасывания дубликатов берут на себя конечные узлы.

ПРИМЕЧАНИЕ: Некоторые параметры логического соединения могут рассматриваться еще и как признаки информационного потока между узлами, установившими это логическое соединение.

Механизм установления логических соединений позволяет реализовывать дифференцированное обслуживание информационных потоков. Разное обслуживание могут получить даже потоки, относящиеся к одной и той же паре конечных узлов. Например, пара конечных узлов может установить два параллельно работающих логических соединения, водном из которых передавать данные в зашифрованном виде, а в другом — открытым текстом.

Передача с установлением соединения предоставляет больше возможностей в плане надежности и безопасности обмена данными, чем дейтаграммная передача. Однако этот способ более медленный, так как он подразумевает дополнительные вычислительные затраты на установление и поддержание логического соединения.

Следующий способ продвижения данных основан на частном случае логического соединения, в число параметров которого входит жестко определенный для всех пакетов маршрут. То есть все пакеты, передаваемые в рамках данного соединения, должны проходить по одному и тому же закрепленному за этим соединением пути.

Единственный заранее проложенный фиксированный маршрут, соединяющий конечные узлы в сети с коммутацией пакетов, называют виртуальным каналом (virtual circuit или virtual channel).

Виртуальные каналы прокладываются для устойчивых информационных потоков. С целью выделения потока данных из общего трафика каждый пакет этого потока помечается специальным видом признака — меткой.

Так же как в сетях с установлением логических соединений, прокладка виртуального канала начинается с отправки из узла-источника специального пакета — запроса на установление соединения. В запросе указываются адрес назначения и метка потока, для которого прокладывается этот виртуальный канал. Запрос, проходя по сети, формирует новую запись в каждом из коммутаторов, расположенных на пути от отправителя до получателя. Запись говорит о том, каким образом коммутатор должен обслуживать пакет, имеющий заданную метку. Образованный виртуальный канал идентифицируется той же меткой. После прокладки виртуального канала сеть может передавать по нему соответствующий поток данных. Во всех пакетах, которые переносят пользовательские данные, адрес назначения уже не указывается, его роль играет метка виртуального канала. При поступлении пакета на входной интерфейс коммутатор читает значение метки из заголовка пришедшего пакета и просматривает свою таблицу коммутации, по которой определяет, на какой выходной порт передать пришедший пакет.

Эта метка в различных технологиях называется по-разному: номер логического канала (Logical Channel Number, LCN) в технологии X.25, идентификатор соединения уровня канала данных (Data Link Connection Identifier, DLCI) в технологии Frame Relay, идентификатор виртуального канала (Virtual Channel Identifier, VCI) в технологии ATM.

Таблица коммутации в сетях, использующих виртуальные каналы, отличается от таблицы коммутации в дейтаграммных сетях. Она содержит записи только о проходящих через коммутатор виртуальных каналах, а не обо всех возможных адресах назначения, как это имеет место в сетях с дейтаграммным алгоритмом продвижения.

Обычно в крупной сети количество проложенных через узел виртуальных каналов существенно меньше общего количества узлов, поэтому и таблицы коммутации в этом случае намного короче, а следовательно, анализ такой таблицы занимает у коммутатора меньше времени. По той же причине метка короче адреса конечного узла, и заголовок пакета в сетях с виртуальными каналами переносит по сети вместо длинного адреса компактный идентификатор потока.

ПРИМЕЧАНИЕ: Использование в сетях техники виртуальных каналов не делает их сетями с коммутацией каналов. Хотя в подобных сетях применяется процедура предварительного установления канала, этот канал является виртуальным, то есть по нему передаются отдельные пакеты, а не потоки информации с постоянной скоростью, как в сетях с коммутацией каналов.

В одной и той же сетевой технологии могут быть задействованы разные способы продвижения данных. Так, дейтаграммный протокол IP используется для передачи данных между отдельными сетями, составляющими Интернет. В то же время обеспечением надежной поставки данных между конечными узлами этой сети занимается протокол TCP, устанавливающий логические соединения без фиксации маршрута. И наконец, Интернет — это пример сети, применяющей технику виртуальных каналов, так как в состав Интернета входит немало сетей ATM и Frame Relay, поддерживающих виртуальные каналы.

Прежде чем проводить техническое сравнение сетей с коммутацией пакетов и сетей коммутацией каналов, проведем их неформальное сравнение на основе продуктивной транспортной аналогии.

Для начала убедимся, что движение на дорогах имеет много общего с перемещением пактов в сети с коммутацией пакетов.

Пусть автомобили в этой аналогии соответствуют пакетам, дороги — каналам связи, а перекрестки — коммутаторам. Подобно пакетам, автомобили перемещаются независимо друг от друга, разделяя пропускную способность дорог и создавая препятствия друг другу. Слишком интенсивный трафик, не соответствующий пропускной способности дороги, приводит к перегруженности дорог, в результате автомобили стоят в пробках, что соответствует очередям пакетов в коммутаторах.

На перекрестках происходит «коммутация» потоков автомобилей, каждый из автомобилей выбирает подходящее направление перекрестка, чтобы попасть в пункт назначения. Конечно, перекресток играет намного более пассивную роль по сравнению с коммутатором пакетов. Его активное участие в обработке трафика можно заметить только на регулируемых перекрестках, где светофор определяет очередность пересечения перекрестка потоками автомобилей. Еще активней, естественно, поведение регулировщика трафика, который может выбрать для продвижения не только поток автомобилей в целом, но и отдельный автомобиль.

Как и в сетях с коммутацией пакетов, к образованию заторов на дорогах приводит неравномерность движения автомобилей. Так, даже кратковременное снижение скорости одного автомобиля на узкой дороге может создать большую пробку, которой бы не было, если бы все автомобили всегда двигались с одной и той же скоростью и равными интервалами.

А теперь попробуем найти общее в автомобильном движении и в сетях с коммутацией каналов.

Иногда на дороге возникает ситуация, когда нужно обеспечить особые условия для движения колонны автомобилей. Например, представим, что очень длинная

колонна автобусов перевозит детей из города в летний лагерь по многополосному шоссе. Для того чтобы колонна двигалась без препятствий, заранее для ее движения разрабатывается маршрут.

Затем на протяжении всего этого маршрута, который пересекает несколько перекрестков, для колонны выделяется отдельная полоса на всех отрезках шоссе. При этом полоса освобождается от другого трафика еще за некоторое время до начала движения колонны, и это резервирование отменяется только после того, как колонна достигает пункта назначения.

Во время движения все автомобили колонны едут с одинаковой скоростью и приблизительно равными интервалами между собой, не создавая препятствий друг другу. Очевидно, что для колонны автомобилей создаются наиболее благоприятные условия движения, но при этом автомобили теряют свою самостоятельность, превращаясь в поток, из которого нельзя «свернуть» в сторону. Дорога при такой организации движения используется не рационально, так как полоса простаивает значительную часть времени, как и полоса пропускания в сетях с коммутацией каналов.

Вернемся от автомобилей к сетевому трафику. Пусть пользователю сети необходимо передать достаточно неравномерный трафик, состоящий из периодов активности и пауз. Представим также, что он может выбрать, через какую сеть, с коммутацией каналов или пакетов, передавать свой трафик, причем в обеих сетях производительность каналов связи одинаковы. Очевидно, что более эффективной с точки зрения временных затрат для нашего пользователя была бы работа в сети с коммутацией каналов, где ему в единоличное владение предоставляется зарезервированный канал связи. При этом способе все данные поступали бы адресату без задержки.

Тот факт, что значительную часть времени зарезервированный канал будет простаивать (во время пауз), нашего пользователя не волнует — ему важно быстро решить собственную задачу.

Если бы пользователь обратился к услугам сети с коммутацией пакетов, то процесс передачи данных оказался бы более медленным, так как его пакеты, вероятно, не раз задерживались бы в очередях, ожидая освобождения необходимых сетевых ресурсов наравне с пакетами других абонентов.

Лекция 4. Архитектура и стандартизация сетей. Модель ISO OSI.

Архитектура подразумевает представление сети в виде системы элементов, каждый из которых выполняет определенную частную функцию, при этом все элементы вместе согласованно решают общую задачу взаимодействия компьютеров. Другими словами, архитектура сети отражает декомпозицию общей задачи взаимодействия компьютеров на отдельные подзадачи, которые должны решаться отдельными элементами сети. Одним из важных элементов архитектуры сети является коммуникационный протокол — формализованный набор правил взаимодействия узлов сети. Прорывом в стандартизации архитектуры компьютерной сети стала разработка модели взаимодействия открытых систем (Open System Interconnection, OSI), которая в начале 80-х годов обобщила накопленный к тому времени опыт. Модель OSI является международным стандартом и определяет 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.

Существующие сегодня (или существовавшие еще недавно) стеки протоколов в целом отражают архитектуру модели OSI. Однако в каждом стеке протоколов имеются свои особенности и отличия от архитектуры OSI. Так, наиболее популярный стек TCP/IP состоит из четырех уровней. Стандартная архитектура компьютерной сети определяет также распределение протоколов между элементами сети — конечными узлами (компьютерами) и промежуточными узлами (коммутаторами и маршрутизаторами). Промежуточные узлы выполняют

только транспортные функции стека протоколов, передавая трафик между конечными узлами. Конечные узлы поддерживают весь стек протоколов, предоставляя информационные услуги, например веб-сервис. Такое распределение функций означает смещение «интеллекта» сети на ее периферию.

Организация взаимодействия между устройствами сети является сложной задачей. Для решения сложных задач используется известный универсальный прием — декомпозиция, то есть разбиение одной сложной задачи на несколько более простых задач-модулей. Декомпозиция состоит в четком определении функций каждого модуля, а также порядка их взаимодействия (то есть межмодульных интерфейсов). При таком подходе каждый модуль можно рассматривать как «черный ящик», абстрагируясь от его внутренних механизмов и концентрируя внимание на способе взаимодействия этих модулей. В результате такого логического упрощения задачи появляется возможность независимого тестирования, разработки и модификации модулей.

Многоуровневый подход

Еще более эффективной концепцией, развивающей идею декомпозиции, является многоуровневый подход. После представления исходной задачи в виде множества модулей эти модули группируют и упорядочивают по уровням, образующим иерархию. В соответствии с принципом иерархии для каждого промежуточного уровня можно указать непосредственно примыкающие к нему соседние вышележащий и нижележащий уровни.

С одной стороны, группа модулей, составляющих каждый уровень, для решения своих задач должна обращаться с запросами только к модулям соседнего нижележащего уровня. С другой стороны, результаты работы каждого из модулей, отнесенных к некоторому

уровню, могут быть переданы только модулям соседнего вышележащего уровня. Такая иерархическая декомпозиция задачи предполагает четкое определение функций и интерфейсов не только отдельных модулей, но и каждого уровня.

Межуровневый интерфейс, называемый также интерфейсом услуг, определяет набор функций, которые нижележащий уровень предоставляет вышележащему.

Такой подход дает возможность проводить разработку, тестирование и модификацию отдельного уровня независимо от других уровней. Иерархическая декомпозиция позволяет, двигаясь от более низкого уровня к более высокому, переходить ко все более и более абстрактному, а значит, более простому представлению исходной задачи.

Задача организации взаимодействия компьютеров в сети тоже может быть представлена в виде иерархически организованного множества модулей. Например, модулям нижнего уровня можно поручить вопросы, связанные с надежной передачей информации между двумя соседними узлами, а модулям следующего, более высокого уровня — транспортировку сообщений в пределах всей сети. Очевидно, что последняя задача — организация связи двух любых, не обязательно соседних, узлов — является более общей и поэтому ее решение может быть получено путем многократных обращений к нижележащему уровню. Так, организация взаимодействия узлов А и В может быть сведена к поочередному взаимодействию пар промежуточных смежных узлов.

Протокол и стек протоколов

В сущности, термины «протокол» и «интерфейс» выражают одно и то же понятие — формализованное описание процедуры взаимодействия двух объектов, но традиционно 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, протоколы верхних уровней, как правило, программными средствами.

Многоуровневое представление средств сетевого взаимодействия имеет свою специфику, связанную с тем, что в процессе обмена сообщениями участвуют, но меньшей мере, две стороны, то есть в данном случае необходимо организовать согласованную работу двух иерархий аппаратных и программных средств на разных компьютерах. Оба участника сетевого обмени должны принять множество соглашений. Например, они должны согласовать уровни и форму электрических сигналов, способ определения размера сообщений, договориться о методах контроля достоверности и т. п. Другими словами, соглашения должны быть приняты на всех уровнях, начиная от самого низкого — уровня передачи битов, и заканчивая самым высоким, реализующим обслуживание пользователей сети.

Программный модуль, реализующий некоторый протокол, называют протокольной сущностью, или, для краткости, тоже протоколом. Понятно, что один и тот же протокол может быть реализован с разной степенью эффективности. Именно поэтому при сравнении протоколов следует учитывать не только логику их работы, но и качество программной реализации. Более того, на эффективность взаимодействия устройств в сети

влияет качество всей совокупности протоколов, составляющих стек, в частности то, насколько рационально распределены функции между протоколами разных уровней и насколько хорошо определены интерфейсы между ними.

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 интерпретации заголовков поступающих к нему сообщений и выполнении связанных с этим действий. Заголовки сообщений разных протоколов имеют разную структуру, что соответствует различиям в их функциональности. Понятно, что чем сложнее структура заголовка сообщения, тем более сложные функции возложены на соответствующий протокол.

Модель OSI

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

В начале 80-х годов ряд международных организаций по стандартизации, в частности International Organization for Standardization (ISO), часто называемая International Standards Organization, а также International Telecommunications Union (ITU) и некоторые другие,

разработали стандартную модель взаимодействия открытых систем (Open System Interconnection, OSI). Эта модель сыграла значительную роль в развитии компьютерных сетей.

К концу 70-х годов в мире уже существовало большое количество фирменных стеков коммуникационных протоколов, среди которых можно назвать, например, такие популярные стеки, как DECnet, TCP/IP и SNA. Подобное разнообразие средств межсетевого взаимодействия вывело на первый план проблему несовместимости устройств, использующих разные протоколы. Одним из путей разрешения этой проблемы в то время виделся всеобщий переход на единый, общий для всех систем стек протоколов, созданный с учетом недостатков уже существующих стеков. Такой академический подход к созданию нового стека начался с разработки модели OSI и занял семь лет (с 1977 по 1984 год). Назначение модели OSI состоит в обобщенном представлении средств сетевого взаимодействия. Она разрабатывалась в качестве своего рода универсального языка сетевых специалистов, именно поэтому ее называют справочной моделью.

Модель OSI определяет, во-первых, уровни взаимодействия систем в сетях с коммутацией пакетов, во-вторых, стандартные названия уровней, в-третьих, функции, которые должен выполнять каждый уровень. Модель OSI не содержит описаний реализаций конкретного набора протоколов.

В модели OSI средства взаимодействия делятся на семь уровней: прикладной, представления, сеансовый, транспортный, сетевой, канальный и физический. Каждый уро-вень имеет дело с совершенно определенным аспектом взаимодействия сетевых устройств.

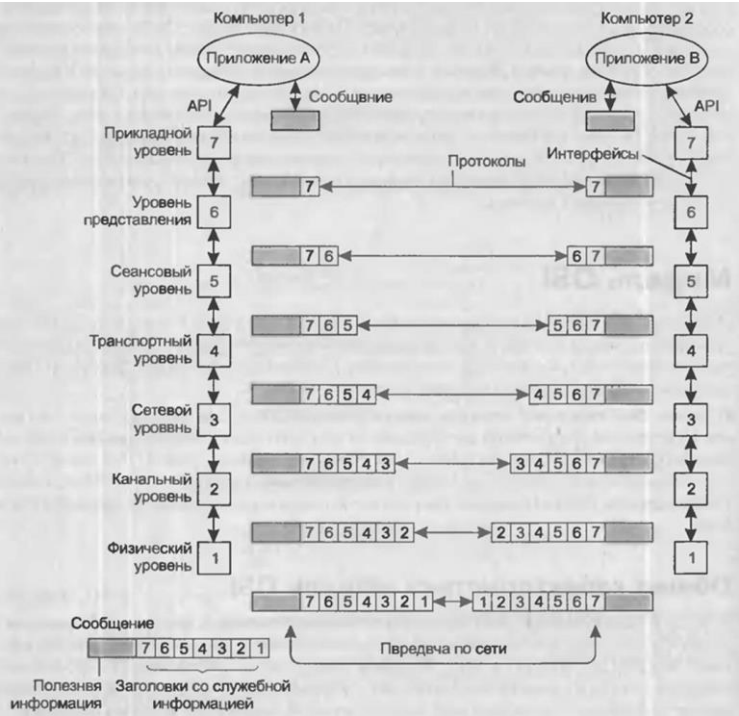


Рисунок 4.1 — Модель взаимодействия открытых систем ISO / OSI [1].

Модель OSI описывает только системные средства взаимодействия, реализуемые операционной системой, системными утилитами, системными аппаратными средствами. Модель не включает средства взаимодействия приложений конечных пользователей. Важно различать уровень взаимодействия приложений и прикладной уровень семиуровневой модели.

Приложения могут реализовывать собственные протоколы взаимодействия, используя для этих целей многоуровневую совокупность системных средств. Именно для этого в распоряжение программистов предоставляется прикладной программный интерфейс (Application Program Interface, API). В соответствии с

идеальной схемой модели OSI приложение может обращаться с запросами только к самому верхнему уровню — прикладному, однако на практике многие стеки коммуникационных протоколов предоставляют возможность программистам напрямую обращаться к сервисам, или службам, расположенных ниже уровней. Например, некоторые СУБД имеют встроенные средства удаленного доступа к файлам. В этом случае приложение, выполняя доступ к удаленным ресурсам, не использует системную файловую службу; оно обходит верхние уровни модели OSI и обращается непосредственно к ответственным за транспортировку сообщений по сети системным средствам, которые располагаются на нижних уровнях модели OSI.

Итак, пусть приложение узла А хочет взаимодействовать с приложением узла В. Для этого приложение А обращается с запросом к прикладному уровню, например к файловой службе. На основании этого запроса программное обеспечение прикладного уровня формирует сообщение стандартного формата. Но для того чтобы доставить эту информацию по назначению, предстоит решить еще много задач, ответственность за которые несут нижележащие уровни.

После формирования сообщения прикладной уровень направляет его вниз по стеку уровню представления. Протокол уровня представления на основании информации, полученной из заголовка сообщения прикладного уровня, выполняет требуемые действия и добавляет к сообщению собственную служебную информацию — заголовок уровня представления, в котором содержатся указания для протокола уровня представления машины-адресата. Полученное в результате сообщение передается вниз сеансовому уровню, который, в свою очередь, добавляет

свой заголовок и т. д. (Некоторые реализации протоколов помещают служебную информацию не только в начале сообщения в виде заголовка, но и в конце в виде так называемого концевика). Наконец, сообщение достигает нижнего, физического, уровня, который собственно и передает его по линиям связи машине-адресату. К этому моменту сообщение «обрастает» заголовками всех уровней (рис. 4.2).

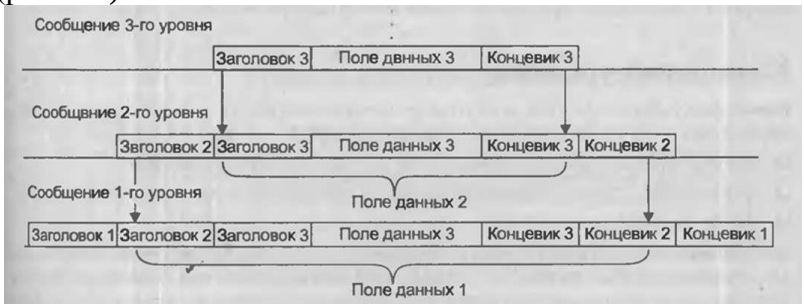


Рис. 4.2 — Вложенность сообщений различных уровней.

Физический уровень помещает сообщение на физический выходной интерфейс компьютера 1, и оно начинает свое «путешествие» по сети (до этого момента сообщение передавалось от одного уровню другому в пределах компьютера 1).

Когда сообщение по сети поступает на входной интерфейс компьютера 2, оно принимается его физическим уровнем и последовательно перемещается вверх с уровня на уровень. Каждый уровень анализирует и обрабатывает заголовок своего уровня, выполняя соответствующие функции, а затем удаляет этот заголовок и передает сообщение вышележащему уровню.

Как видно из описания, протокольные сущности одного уровня не общаются между собой непосредственно, в этом общении всегда участвуют посредники — средства протоколов нижележащих уровней. И только физические

уровни различных узлов взаимодействуют непосредственно.

В стандартах ISO для обозначения единиц обмена данными, с которыми имеют дело протоколы разных уровней, используется общее название протокольная единица данных (Protocol Data Unit, PDU). Для обозначения единиц обмена данными конкретных уровней часто используются специальные названия, в частности: сообщение, кадр, пакет, дейтаграмма, сегмент.

Физический уровень

Физический уровень (physical layer) имеет, дело с передачей потока битов по физическим каналам связи, таким как коаксиальный кабель, витая пара, оптоволоконный кабель или цифровой территориальный канал. Функции физического уровня реализуются на всех устройствах, подключенных к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером или последовательным портом.

Примером протокола физического уровня может служить спецификация 10Base-T технологии Ethernet, которая определяет в качестве используемого кабеля неэкранированную витую пару категории 3 с волновым сопротивлением 100 Ом, разъем RJ-45, максимальную длину физического сегмента 100 метров, манчестерский код для представления данных в кабеле, а также некоторые другие характеристики среды и электрических сигналов. Физический уровень не вникает в смысл информации, которую он передает. Для него эта информация представляет собой однородный поток битов, которые нужно доставить без искажений и в соответствии с заданной тактовой частотой (интервалом между соседними битами).

Канальный уровень

Канальный уровень (data link layer) обеспечивает прозрачность соединения для сетевого уровня. Для этого он предлагает ему следующие услуги:

- установление логического соединения между взаимодействующими узлами;
- согласование в рамках соединения скоростей передатчика и приемника информации;
- обеспечение надежной передачи, обнаружение и коррекция ошибок.

Для решения этих задач канальный уровень формирует из пакетов собственные протокольные единицы данных — кадры, состоящие из поля данных и заголовка. Канальный уровень помещает пакет в поле данных одного или нескольких кадров и заполняет собственной служебной информацией заголовок кадра.

В сетях, построенных на основе разделяемой среды, физический уровень выполняет еще одну функцию — проверяет доступность разделяемой среды. Эту функцию иногда выделяют в отдельный подуровень управления доступом к среде (Medium Access Control, MAC).

Протоколы канального уровня реализуются как на конечных узлах (средствами сетевых адаптеров и их драйверов), так и на всех промежуточных сетевых устройствах.

Рассмотрим более подробно работу канального уровня, начиная с момента, когда сетевой уровень отправителя передает канальному уровню пакет, а также указание, какому узлу его передать. Для решения этой задачи канальный уровень создает кадр, который имеет поле данных и заголовок. Канальный уровень помешает (инкапсулирует) пакет в поле данных кадра н заполняет соответствующей служебной информацией заголовок кадра. Важнейшей информацией заголовка кадра является

адрес назначения, на основании которого коммутаторы сети будут продвигать пакет.

Одной из задач канального уровня является обнаружение и коррекция ошибок. Канальный уровень может обеспечить надежность передачи, например, путем фиксирования границ кадра, помещая специальную последовательность битов в его начало и конец, а затем добавляя к кадру контрольную сумму. Контрольная сумма вычисляется по некоторому алгоритму как функция от всех байтов кадра. На стороне получателя канальный уровень группирует биты, поступающие с физического уровня, в кадры, снова вычисляет контрольную сумму полученных данных и сравнивает результат с контрольной суммой, переданной в кадре. Если они совпадают, кадр считается правильным. Если же контрольные суммы не совпадают, фиксируется ошибка.

В функции канального уровня входит не только обнаружение ошибок, но и их исправление за счет повторной передачи поврежденных кадров. Однако эта функция не является обязательной и в некоторых реализациях канального уровня она отсутствует, например, в Ethernet.

Прежде чем переправить кадр физическому уровню для непосредственной передачи данных в сеть, канальному уровню может потребоваться решить еще одну важную задачу. Если в сети используется разделяемая среда, то прежде чем физический уровень начнет передавать данные, канальный уровень должен проверить доступность среды. Функции проверки доступности разделяемой среды иногда выделяют в отдельный подуровень управления доступом к среде (подуровень MAC). Если разделяемая среда освободилась (когда она не используется, то такая проверка, конечно, пропускается), кадр передается средствами физического уровня в сеть, проходит по каналу

связи и поступает в виде последовательности битов в распоряжение физического уровня узла назначения. Этот уровень в свою очередь передаст полученные биты «наверх» канальному уровню своего узла.

Протокол канального уровня обычно работает в пределах сети, являющейся одной из составляющих более крупной составной сети, объединенной протоколами сетевого уровня. Адреса, с которыми работает протокол канального уровня, используются для доставки кадров только в пределах этой сети, а для перемещения пакетов между сетями применяются уже адреса следующего, сетевого, уровня.

В локальных сетях канальный уровень поддерживает весьма мощный и законченный набор функций по пересылке сообщений между узлами сети. В некоторых случаях протоколы канального уровня локальных сетей оказываются самодостаточными транспортными средствами и могут допускать работу непосредственно поверх себя протоколов прикладного уровня или приложений без привлечения средств сетевого и транспортного уровней. Тем не менее для качественной передачи сообщений в сетях с произвольной топологией функций канального уровня оказывается недостаточно.

Сетевой уровень

Сетевой уровень (network layer) служит для образования единой транспортной системы, объединяющей несколько сетей, называемой составной сетью, или интернетом.

Технология, позволяющая соединять в единую сеть множество сетей, в общем случае построенных на основе разных технологий, называется технологией межсетевого взаимодействия (internetworking)

Если есть несколько сетей, каждая из которых использует собственную техно-логию канального уровня:

Ethernet, FDDI, Token Ring, ATM, Frame Relay. На базе этих технологий любая из указанных сетей может связывать между собой любых пользователей, но только своей сети, и не способна обеспечить передачу данных в другую сеть. Причина такого положения вещей очевидна и кроется в существенных отличиях одной технологии от другой. Даже наиболее близкие технологии LAN — Ethernet, FDDI, Token Ring, — имеющие одну и ту же систему адресации (адреса подуровня MAC, называемые MAC-адресами), отличаются друг от друга форматом используемых кадров и логикой работы протоколов. Еще больше отличий между технологиями LAN и WAN. Во многих технологиях WAN задействована техника предварительно устанавливаемых виртуальных каналов, идентификаторы которых применяются в качестве адресов. Все технологии имеют собственные форматы кадров (в технологии ATM кадр даже называется иначе — ячейкой) и. конечно, собственные стеки протоколов.

Не следует путать интернет (со строчной буквы) с Интернетом (с прописной буквы). Интернет — это самая известная и охватывающая весь мир реализация составной сети, построенная на основе технологии TCP/IP.

Чтобы связать между собой сети, построенные на основе столь отличающихся технологий, нужны дополнительные средства, и такие средства предоставляет сетевой уровень.

Функции сетевого уровня реализуются:

- группой протоколов;
- специальными устройствами — маршрутизаторами.

Одной из функций маршрутизатора является физическое соединение сетей. Маршрутизатор имеет несколько сетевых интерфейсов, подобных интерфейсам компьютера, к каждому из которых может быть

подключена одна сеть. Таким образом, все интерфейсы маршрутизатора можно считать узлами разных сетей. Маршрутизатор может быть реализован программно на базе универсального компьютера (например, типовая конфигурация Unix или Windows включает программный модуль маршрутизатора). Однако чаще маршрутизаторы реализуются на базе специализированных аппаратных платформ. В состав программного обеспечения маршрутизатора входят протокольные модули сетевого уровня.

Итак, чтобы связать сети, необходимо соединить все эти сети маршрутизаторами и установить протокольные модули сетевого уровня на все конечные узлы пользователей, которые хотели бы связываться через составную сеть.

Данные, которые необходимо передать через составную сеть, поступают на сетевой уровень от вышележащего транспортного уровня. Эти данные снабжаются заголовком сетевого уровня. Данные вместе с заголовком образуют пакет — так называется PDU сетевого уровня. Заголовок пакета сетевого уровня имеет унифицированный формат, не зависящий от форматов кадров канального уровня тех сетей, которые могут входить в составную сеть, и несет, наряду с другой служебной информацией, данные об адресе назначения этого пакета.

Для того чтобы протоколы сетевого уровня могли доставлять пакеты любому узлу составной сети, эти узлы должны иметь адреса, уникальные в пределах данной составной сети. Такие адреса называются сетевыми, или глобальными. Каждый узел составной сети, который намерен обмениваться данными с другими узлами составной сети, наряду с адресом, назначенным ему на канальном уровне, должен иметь сетевой адрес. В пакете в

качестве адреса назначения должен быть указан адрес сетевого уровня, на основании которого определяется маршрут пакета.

Определение маршрута является важной задачей сетевого уровня. Маршрут описывается последовательностью сетей (или маршрутизаторов), через которые должен пройти пакет, чтобы попасть к адресату. Маршрутизатор собирает информацию о топологии связей между сетями и на основе этой информации строит таблицы коммутации, которые в данном случае носят специальное название таблиц маршрутизации.

В соответствии с многоуровневым подходом сетевой уровень для решения своей задачи обращается к нижележащему канальному уровню. Весь путь через составную сеть разбивается на участки от одного маршрутизатора до другого, причем каждый участок соответствует пути через отдельную сеть.

Для того чтобы передать пакет через очередную сеть, сетевой уровень помещает его в поле данных кадра соответствующей канальной технологии, указывая в заголовке кадра канальный адрес интерфейса следующего маршрутизатора. Сеть, используя свою канальную технологию, доставляет кадр с инкапсулированным в него пакетом по заданному адресу. Маршрутизатор извлекает пакет из прибывшего кадра и после необходимой обработки передает пакет для дальнейшей транспортировки в следующую сеть, предварительно упаковав его в новый кадр канального уровня в общем случае другой технологии. Таким образом, сетевой уровень играет роль координатора, организующего совместную работу сетей, построенных на основе разных технологий.

В общем случае функции сетевого уровня шире, чем обеспечение обмена в пределах составной сети. Так,

сетевой уровень решает задачу создания надежных и гибких барьеров на пути нежелательного трафика между сетями.

В заключение отметим, что на сетевом уровне определяются два вида протоколов. Первый вид маршрутизируемые протоколы — реализуют продвижение пакетов через сеть. Именно эти протоколы обычно имеют в виду, когда говорят о протоколах сетевого уровня. Однако часто к сетевому уровню относят и другой вид протоколов, называемых маршрутизирующими протоколами, или протоколами маршрутизации. С помощью этих протоколов маршрутизаторы собирают информацию о топологии межсетевых соединений, на основании которой осуществляется выбор маршрута продвижения пакетов.

Транспортный уровень

На пути от отправителя к получателю пакеты могут быть искажены или утеряны. Хотя некоторые приложения имеют собственные средства обработки ошибок, существуют и такие, которые предпочитают сразу иметь дело с надежным соединением.

Транспортный уровень (transport layer) обеспечивает приложениям или верхним уровням стека — прикладному, представления и сеансовому — передачу данных с той степенью надежности, которая им требуется. Модель OSI определяет пять классов транспортного сервиса от низшего класса 0 до высшего класса 4. Эти виды сервиса отличаются качеством предоставляемых услуг: срочностью, возможностью восстановления прерванной связи, наличием средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол, а главное — способностью к обнаружению и исправлению

ошибок передачи, таких как искажение, потеря и дублирование пакетов.

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

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

Протоколы нижних четырех уровней обобщенно называют сетевым транспортом, или транспортной

подсистемой, так как они полностью решают задачу транспортировки сообщений с заданным уровнем качества в составных сетях с произвольной топологией и различными технологиями. Оставшиеся три верхних уровня решают задачи предоставления прикладных сервисов, используя нижележащую транспортную подсистему.

Сеансовый уровень

Сеансовый уровень (session layer) управляет взаимодействием сторон: фиксирует, какая из сторон является активной в настоящий момент, и предоставляет средства синхронизации сеанса. Эти средства позволяют в ходе длинных передач сохранять информацию о состоянии этих передач в виде контрольных точек, чтобы в случае отказа можно было вернуться назад к последней контрольной точке, а не начинать все с начала. На практике немногие приложения используют сеансовый уровень, и он редко реализуется в виде отдельных протоколов. Функции этого уровня часто объединяют с функциями прикладного уровня и реализуют в одном протоколе.

Уровень представления

Уровень представления (presentation layer), как явствует из его названия, обеспечивает представление передаваемой по сети информации, не меняя при этом ее содержания. За счет уровня представления информация, передаваемая прикладным уровнем одной системы, всегда понятна прикладному уровню другой системы. С помощью средств данного уровня протоколы прикладных уровней могут преодолеть синтаксические различия в представлении данных или же различия в кодах символов, например кодов ASCII и EBCDIC. На этом уровне могут выполняться шифрование и дешифрирование данных, благодаря которым секретность обмена данными

обеспечивается сразу для всех прикладных служб. Примером такого протокола является протокол SSL (Secure Socket Layer — слой защищенных сокетов), который обеспечивает секретный обмен сообщениями для протоколов прикладного уровня стека TCP/IP.

Прикладной уровень

Прикладной уровень (application layer) — это в действительности просто набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые веб-страницы, а также организуют свою совместную работу, например, по протоколу электронной почты. Единица данных, которой оперирует прикладной уровень, обычно называется сообщением. Существует очень большое разнообразие протоколов и соответствующих служб прикладного уровня. К протоколам прикладного уровня относится, в частности, упоминавшийся ранее протокол HTTP, с помощью которого браузер взаимодействует с веб-сервером. Приведем в качестве примера также несколько наиболее распространенных реализаций сетевых файловых служб: NFS и FTP в стеке TCP/IP, SMB в Microsoft Windows, NCP в операционной системе Novell NetWare.

Лекция 5. Модель OSI и сети с коммутацией каналов. Стеки коммуникационных технологий

Как уже было упомянуто, модель OSI описывает процесс взаимодействия устройств в сети с коммутацией пакетов. А как же обстоит дело с сетями коммутации каналов? Существует ли для них собственная справочная модель? Можно ли сопоставить функции технологий коммутации каналов с уровнями модели OSI?

Да, для представления структуры средств межсетевого взаимодействия сетей с коммутацией каналов также используется многоуровневый подход, в соответствии с которым существуют протоколы нескольких уровней, образующих иерархию. Однако общей справочной модели, подобной модели OSI, для сетей с коммутацией каналов не существует. Например, различные типы телефонных сетей имеют собственные стеки протоколов, отличающиеся количеством уровней и распределением функций между уровнями. Первичные сети, такие как SDH или DWDM, также обладают собственной иерархией протоколов. Ситуация усложняется еще и тем, что практически все типы современных сетей с коммутацией каналов задействуют эту технику только для передачи пользовательских данных, а для управления процессом установления соединений в сети и общего управления сетью применяют технику коммутации пакетов. Такими сетями являются, например, сети ISDN, SDH, DWDM.

Для сетей с коммутацией пакетов сети с коммутацией каналов предоставляют сервис физического уровня, хотя сами они устроены достаточно сложно и поддерживают собственную иерархию протоколов.

Рассмотрим, к примеру, случай, когда несколько локальных пакетных сетей связываются между собой через цифровую телефонную сеть. Очевидно, что функции

создания составной сети выполняют протоколы сетевого уровня, поэтому мы устанавливаем в каждой локальной сети маршрутизатор. Маршрутизатор должен быть оснащен интерфейсом, способным установить соединение через телефонную сеть с другой локальной сетью. После того как такое соединение установлено, в телефонной сети образуется ноток битов, передаваемых с постоянной скоростью. Это соединение и предоставляет маршрутизаторам сервис физического уровня. Для того чтобы организовать передачу данных, маршрутизаторы используют поверх этого физического канала какой-либо двухточечный протокол канального уровня.

Универсальный тезис о пользе стандартизации, справедливый для всех отраслей, в компьютерных сетях приобретает особое значение. Суть сети — это соединение разного оборудования, а значит, проблема совместимости является здесь одной из наиболее острых. Без согласования всеми производителями общепринятых стандартов для оборудования и протоколов прогресс в деле «строительства» сетей был бы невозможен. Поэтому все развитие компьютерной отрасли, в конечном счете, 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 рассмотренная ранее модель взаимодействия открытых систем (OSI).

Понятие открытой системы

Открытой может быть названа любая система (компьютер, вычислительная сеть, ОС, программный пакет, другие аппаратные и программные продукты), которая построена в соответствии с открытыми спецификациями.

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

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

Открытый характер стандартов и спецификаций важен не только для коммуникационных протоколов, но и для разнообразных устройств и программ, выпускаемых для построения сети. Нужно отметить, что большинство стандартов, принимаемых сегодня, носят открытый характер. Время закрытых систем, точные спецификации на которые были известны только фирме-производителю, ушло. Все осознали, что возможность взаимодействия с продуктами конкурентов не снижает, а наоборот, повышает ценность изделия, так как позволяет применять его в большем количестве работающих сетей, собранных из продуктов разных производителей. Поэтому даже такие фирмы, как IBM, Novell и Microsoft, ранее выпускавшие закрытые системы, сегодня активно участвуют в разработке открытых стандартов и применяют их в своих продуктах.

Для реальных систем полная открытость является недостижимым идеалом. Как правило, даже в системах, называемых открытыми, этому определению соответствуют лишь некоторые части, поддерживающие внешние интерфейсы. Например, открытость семейства операционных систем Unix заключается, помимо всего прочего, в наличии стандартизованного программного интерфейса между ядром и приложениями, что позволяет легко переносить приложения из среды одной версии Unix в среду другой версии.

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

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

- возможность построения сети из аппаратных и программных средств различных производителей, придерживающихся одного и того же стандарта;
- безболезненная замена отдельных компонентов сети другими, более совершенными, что позволяет сети развиваться с минимальными затратами;
- легкость сопряжения одной сети с другой.

Источники стандартов

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

- стандарты отдельных фирм, например стек протоколов SNA компании IBM или графический

интерфейс OPEN LOOK для Unix-систем компании Sun;

- стандарты специальных комитетов и объединений создаются несколькими компаниями, например стандарты технологии ATM, разрабатываемые специально созданным объединением ATM Forum, которое насчитывает около 100 коллективных участников, или стандарты союза Fast Ethernet Alliance, касающиеся технологии 100 Мбит Ethernet;
- национальные стандарты, например стандарт FDDI, представляющий один из многочисленных стандартов института ANSI, или стандарты безопасности для операционных систем, разработанные центром NCSC Министерства обороны США;
- международные стандарты, например модель и стек коммуникационных протоколов Международной организации по стандартизации (ISO), многочисленные стандарты Международного союза электросвязи (ITU), в том числе стандарты на сети с коммутацией пакетов X.25, сети Frame Relay, ISDN, модемы и многие другие.

Некоторые стандарты, непрерывно развиваясь, могут переходить из одной категории в другую. В частности, фирменные стандарты на продукцию, получившую широкое распространение, обычно становятся международными стандартами де-факто, так как вынуждают производителей из разных стран следовать фирменным стандартам, чтобы обеспечить совместимость своих изделий с этими популярными продуктами.

Более того, ввиду широкого распространения некоторые фирменные стандарты становятся основой для

национальных и международных стандартов де-юре. Например, стандарт Ethernet, первоначально разработанный компаниями Digital Equipment, Intel и Xerox, через некоторое время и в несколько измененном виде был принят как национальный стандарт IEEE 802.3, а затем организация ISO утвердила его в качестве международного стандарта ISO 8802.3.

Ярким примером открытой системы является Интернет. Эта международная сеть развивалась в полном соответствии с требованиями, предъявляемыми к открытым системам. В разработке ее стандартов принимали участие тысячи специалистов-пользователей этой сети из различных университетов, научных организаций и фирм-производителей вычислительной аппаратуры и программного обеспечения, работающих в разных странах. Само название стандартов, определяющих работу Интернета, — темы для обсуждения (Request For Comments, RFC) — показывает гласный и открытый характер принимаемых стандартов. В результате Интернет сумел объединить в себе разнообразное оборудование и программное обеспечение огромного числа сетей, разбросанных по всему миру. Ввиду постоянной растущей популярности Интернета документы RFC становятся международными стандартами де-факто, многие из которых затем приобретают статус официальных международных стандартов в результате их утверждения какой-либо организацией по стандартизации, как правило, ISO и ITU-T.

Существует несколько организационных подразделений, отвечающих за развитие и, в частности, за стандартизацию архитектуры и протоколов Интернета. Основным из них является научно-административное сообщество Интернета (Internet Society, ISOC), объединяющее около 100 000 человек, которое занимается

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. Именно IETF определяет спецификации, которые затем становятся стандартами Интернета. Процесс разработки и принятия стандарта для протокола Интернета состоит из ряда обязательных этапов, или стадий, включающих непременную экспериментальную проверку.

В соответствии с принципом открытости Интернета все документы RFC, в отличие, скажем, от стандартов ISO, находятся в свободном доступе. Список RFC можно найти, в частности, на сайте www.rfc-editor.org.

Стандартные стеки коммуникационных протоколов

Важнейшим направлением стандартизации в области вычислительных 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).

Стек OSI

Важно различать модель OSI и стек протоколов OSI. В то время как модель OSI является концептуальной схемой взаимодействия открытых систем, стек OSI представляет собой набор спецификаций конкретных протоколов.

В отличие от других стеков протоколов, стек OSI полностью соответствует модели OSI, включая спецификации протоколов для всех семи уровней взаимодействия, определенных в этой модели. Это и понятно, разработчики стека OSI использовали модель OSI как прямое руководство к действию.

Протоколы стека OSI отличает сложность и неоднозначность спецификаций. Эти свойства явились результатом общей политики разработчиков стека, стремившихся учесть в своих протоколах все многообразие уже существующих и появляющихся технологий. На физическом и канальном уровнях стек OSI поддерживает протоколы Ethernet, Token Ring, FDDI, а также протоколы LLC, X.25 и ISDN, то есть использует все разработанные вне стека популярные протоколы нижних уровней, как и большинство других стеков. Сетевой уровень включает сравнительно редко используемые протоколы Connection- I oriented Network Protocol (CONP) и Connectionless Network Protocol (CLNP). Как следует из названий, первый из них ориентирован на соединение (connection-oriented), второй — нет (connectionless).

Более популярны протоколы маршрутизации стека OSI: ES-IS (End System — Intermediate System) между конечной и промежуточной системами и IS-IS (Intermediate System — I Intermediate System) между промежуточными системами.

Транспортный уровень стека OSI в соответствии с функциями, определенными для него в модели OSI, скрывает различия между сетевыми сервисами с установлением соединения и без установления соединения, так что пользователи получают требуемое качество обслуживания независимо от нижележащего сетевого уровня. Чтобы обеспечить это, транспортный уровень

требует, чтобы пользователь задал нужное качество обслуживания.

Службы прикладного уровня обеспечивают передачу файлов, эмуляцию терминала, службу каталогов и почту. Из них наиболее популярными являются служба каталогов (стандарт X.500), электронная почта (X.400), протокол виртуального терминала (VTP), протокол передачи, доступа и управления файлами (FTAM), протокол пересылки и управления работами (JTM).

Стек IPX/SPX

Стек IPX/SPX является оригинальным стеком протоколов фирмы Novell, разработанным для сетевой операционной системы NetWare еще в начале 80-х годов. Название стеку дали протоколы сетевого и транспортного уровней — Internetwork Packet Exchange (IPX) и Sequenced Packet Exchange (SPX). К сетевому уровню этого стека отнесены также протоколы маршрутизации RIP и NLSP. А в качестве представителей трех верхних уровней на рисунке приведены два популярных протокола: протокол удаленного доступа к файлам NetWare Core Protocol (NCP) и протокол объявления о сервисах Service Advertising Protocol (SAP).

До 1996 гада стек IPX/SPX был бесспорным мировым лидером по числу установленных копий, но затем картина резко изменилась — стек TCP/IP по темпам роста числа установок намного стал опережать другие стеки, а с 1998 года вышел в лидеры и в абсолютном выражении.

Многие особенности стека IPX/SPX обусловлены ориентацией ранних версий ОС NetWare на работу в локальных сетях небольших размеров, состоящих из персональных компьютеров со скромными ресурсами. Понятно, что для таких компьютеров компании Novell нужны были протоколы, на реализацию которых

требовалось бы минимальное количество оперативной памяти (ограниченной в IBM-совместимых компьютерах под управлением MS-DOS объемом 640 Кбайт) и которые бы быстро работали на процессорах небольшой вычислительной мощности. В результате протоколы стека IPX/SPX до недавнего времени отлично справлялись с работой в локальных сетях. Однако в крупных корпоративных сетях они слишком перегружали медленные глобальные связи широковещательными пакетами, интенсивно использующимися несколькими протоколами этого стека, например протоколом SAP. Это обстоятельство, а также тот факт, что стек IPX/SPX является собственностью фирмы Novell и на его реализацию нужно получать лицензию (то есть открытые спецификации не поддерживались), долгое время ограничивали распространенность его только сетями NetWare.

Стек NetBIOS/SMB

Стек NetBIOS/SMB является совместной разработкой компаний IBM и Microsoft. На физическом и канальном уровнях этого стека также задействованы уже получившие распространение протоколы, такие как Ethernet, Token Ring, FDDI, а на верхних уровнях — специфические протоколы NetBEUI и SMB.

Протокол Network Basic Input / Output System (NetBIOS) появился в 1984 году как сетевое расширение стандартных функций базовой системы ввода-вывода (BIOS) IBM PC для сетевой программы PC Network фирмы IBM. В дальнейшем этот протокол был заменен так называемым протоколом расширенного пользовательского интерфейса NetBEUI (NetBIOS Extended User Interface). Для совместимости приложений в качестве интерфейса к протоколу NetBEUI был сохранен интерфейс NetBIOS. NetBEUI разрабатывался как эффективный протокол,

потребляющий немного ресурсов и предназначенный для сетей, насчитывающих не более 200 рабочих станций. Этот протокол содержит много полезных сетевых функций, которые можно отнести к транспортному и сеансовому уровням модели OSI, однако с его помощью невозможна маршрутизация пакетов. Это ограничивает применение протокола NetBEUI локальными сетями, не разделенными на подсети, и делает невозможным его использование в составных сетях.

Протокол Server Message Block (SMB) поддерживает функции сеансового уровня, уровня представления и прикладного уровня. На основе SMB реализуется файловая служба, а также службы печати и передачи сообщений между приложениями.

Стек TCP/IP

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

Соответствие популярных стеков протоколов модели OSI

В большинстве случаев разработчики стеков отдавали предпочтение скорости работы сети в ущерб модульности — ни один стек, кроме стека OSI, не разбит на семь уровней. Чаще всего в стеке явно выделяются 3-4

уровня: уровень сетевых адаптеров, в котором реализуются протоколы физического и канального уровней, сетевой уровень, транспортный уровень и уровень служб, вбирающий в себя функции сеансового уровня, уровня представления и прикладного уровня.

Структура стеков протоколов часто не соответствует рекомендуемому моделью OSI разбиению на уровни и по другим причинам. Давайте вспомним, чем характеризуется идеальная многоуровневая декомпозиция. С одной стороны, необходимо соблюсти принцип иерархии: каждый вышележащий уровень обращается с запросами только к нижележащему, а нижележащий предоставляет свои сервисы только непосредственно соседствующему с ним вышележащему. В стеках протоколов это приводит к тому, что PDU вышележащего уровня всегда инкапсулируется в PDU нижележащего.

С другой же стороны, идеальная многоуровневая декомпозиция предполагает, что все модули, отнесенные к одному уровню, ответственны за решение общей для всех них задачи. Однако эти требования часто вступают в противоречие. Например, основной функцией протоколов сетевого уровня стека TCP/IP (так же как и сетевого уровня OSI) является передача пакетов через составную сеть. Для решения этой задачи в стеке TCP/IP предусмотрено несколько протоколов: протокол продвижения IP-пакетов и протоколы маршрутизации RIP, OSPF и др. Если считать признаком принадлежности к одному и тому же уровню общность решаемых задач, то, очевидно, протокол IP и протоколы маршрутизации должны быть отнесены к одному уровню. Вместе с тем, если принять во внимание, что сообщения протокола RIP инкапсулируются в UDP-дейтаграммы, а сообщения протокола OSPF — в IP-пакеты, то, следуя формально принципу иерархической организации стека, OSPF

следовало бы отнести к транспортному, а RIP — к прикладному уровню. На практике же протоколы маршрутизации обычно включают в сетевой уровень.

Лекция 6. Технологии локальных сетей на разделяемой среде

Алгоритм доступа к разделяемой среде является одним из главных факторов, определяющих эффективность совместного использования среды конечными узлами локальной сети. Можно сказать, что алгоритм доступа формирует «облик» технологии, позволяет отличать данную технологию от других. В технологии Ethernet применяется очень простой алгоритм доступа, позволяющий узлу сети передавать данные в те моменты времени, когда он считает, что разделяемая среда свободна. Простота алгоритма доступа определила простоту и низкую стоимость оборудования Ethernet. Негативным атрибутом алгоритма доступа технологии Ethernet являются коллизии, то есть ситуации, когда кадры, передаваемые разными станциями, сталкиваются друг с другом в общей среде. Коллизии снижают эффективность разделяемой среды и придают работе сети непредсказуемый характер. Первоначальный вариант технологии Ethernet был рассчитан на коаксиальный кабель, который использовался всеми узлами сети в качестве общей шины. Переход на кабельные системы на витой паре и концентраторах (хабах) существенно повысил эксплуатационные характеристики сетей Ethernet.

В технологиях Token Ring и FDDI поддерживались более сложные и эффективные алгоритмы доступа к среде, основанные на передаче друг другу токена — специального кадра, разрешающего доступ. Однако чтобы выжить в конкурентной борьбе с Ethernet, этого преимущества оказалось недостаточно.

Стандартная топология и разделяемая среда

Основная цель, которую ставили перед собой разработчики первых локальных сетей во второй половине 70-х годов, заключалась в нахождении простого и

дешевого решения для объединения в вычислительную сеть нескольких десятков компьютеров, находящихся в пределах одного здания. Решение должно было быть недорогим, поскольку компьютеры, объединявшиеся в сеть, были недороги — появившиеся и быстро распространявшиеся тогда мини-компьютеры стоимостью в 10 000 - 20 000 долларов. Количество их в одной организации было небольшим, поэтому предел в несколько десятков компьютеров представлялся вполне достаточным для практически любой локальной сети. Задача связи локальных сетей в глобальные не была первоочередной, поэтому практически все технологии локальных сетей ее игнорировали.

Для упрощения и, соответственно, удешевления аппаратных и программных решений разработчики первых локальных сет остановились на совместном использовании общей среды передели данных.

Этот метод связи компьютеров впервые был опробован при создании радиосети ALOHA Гавайского университета в начале 70-х под руководством Нормана Абрамсона (Norman Abramson). Радиоканал определенного диапазона частот естественным образом является общей средой для всех передатчиков, использующих частоты этого диапазона для кодирования данных. Сеть ALOHA работала по методу случайного доступа, когда каждый узел мог начать передачу пакета в любой момент времени. Если после этого он не дожидался подтверждения приема в течение определенного тайм-аута, он посылал этот пакет снова Общим был радиоканал с несущей частотой 400 МГц и полосой 40 кГц, что обеспечивало передачу данных со скоростью 9600 бит/с.

Немного позже Роберт Меткалф (Robert Metcalfe) повторил идею разделяемой среды уже для проводного варианта технологии LAN. Непрерывный сегмент

коаксиального кабеля стал аналогом общей радиосреды. Все компьютеры присоединялись к этому сегменту кабеля по схеме монтажного ИЛИ, поэтому при передаче сигналов одним из передатчиков все приемники получали один и тот же сигнал, как и при использовании радиоволн.

В технологиях Token Ring и FDDI тот факт, что компьютеры используют разделяемую среду, не так очевиден, как в случае Ethernet. Физическая топология этих сетей — кольцо каждый узел соединяется кабелем с двумя соседними узлами. Однако эти отрезки кабеля также являются разделяемыми, так как в каждый момент времени только один компьютер может задействовать кольцо для передачи своих пакетов.

Простые стандартные топологии физических связей (звезда у коаксиального кабеля Ethernet и кольцо у Token Ring и FDDI) обеспечивают простоту разделения кабельной среды.

Использование разделяемых сред позволяет упростить логику работы узлов сети. Действительно, поскольку в каждый момент времени выполняется только одна передача, отпадает необходимость в буферизации кадров в транзитных узлах и, как, следствие, в самих транзитных узлах. Соответственно, отпадает необходимость в сложных процедурах управления потоком и борьбы с перегрузками.

Основной недостаток разделяемой среды — плохая масштабируемость. Этот недостаток является принципиальным, так как независимо от метода доступа к среде ее пропускная способность делится между всеми узлами сети. Здесь применимо положение теории очередей, как только коэффициент использования общей среды превышает определенный порог, очереди к среде начинают расти нелинейно, и сеть становится практически неработоспособной. Значение порога зависит от метода

доступа. Так, в сетях ALOHA это значение является крайне низким — всего около 18 %, в сетях Ethernet — около 30 %, а в сетях Token Ring и FDDI оно возросло до 60-70 %.

Локальные сети, являясь пакетными сетями, используют принцип временного мультиплексирования, то есть разделяют передающую среду во времени. Алгоритм управления доступом к среде является одной из важнейших характеристик любой технологии LAN, в значительно большей степени определяющей ее облик, чем метод кодирования сигналов или формат кадра. В технологии Ethernet в качестве алгоритма разделения среды применяется метод случайного доступа. И хотя его трудно назвать совершенным — при росте нагрузки полезная пропускная способность сети резко падает — он благодаря своей простоте стал основой успеха технологии Ethernet. Технологии Token Ring и FDDI используют метод маркерного доступа, основанный на передаче от узла к узлу особого кадра — маркера (токена) доступа. При этом только узел, владеющий маркером доступа, имеет право доступа к разделяемому кольцу. Более детерминированный характер доступа технологий Token Ring и FDDI предопределил более эффективное использование разделяемой среды, чем у технологии Ethernet, но одновременно и усложнил оборудование.

Появление мультимедийных приложений с чувствительным к задержкам трафиком привело к попыткам создания метода доступа, учитывающего систему приоритетов трафика и обеспечивающего для него необходимые характеристики QoS. Результатом этих попыток стало создание технологии 100VG-AnyLAN, для которой был характерен достаточно сложный метод доступа к разделяемой среде. Однако эта технология была создана слишком поздно — в середине 90-х годов, когда преимущества и доступность коммутируемых локальных

сетей «отменили» сам принцип разделения среды (в проводных сетях). Отказ от разделяемой среды привел к исчезновению такого важного компонента технологии локальных сетей как метод доступа. В принципе коммутатор локальной сети работает так же, как и обобщенный коммутатор сети с коммутацией пакетов. Поэтому с распространением коммутаторов стали исчезать различия между технологиями локальных сетей, так как в сети, где все связи между узлами являются индивидуальными, и коммутируемая версия Ethernet, и коммутируемая версия Token Ring работают весьма схоже, различаются только форматы кадров этих технологий. Это обстоятельство, возможно, и имел в виду Роберт Меткалф, когда говорил об удачливости Ethernet — работа коммутируемых локальных сетей Etherhet существенно отличается от работы Etherhet на разделяемой среде, так что ее можно считать новой технологией со старым названием. Хотя, с другой стороны, формат кадра Ethernet сохранился, гак что это дает формальный (хотя и несколько условный) повод считать ее той же самой технологией.

Стандартизация протоколов локальных сетей

Каждая из технологий локальных сетей первоначально появлялась как фирменная технология; так, например, технология Ethernet «появилась на свет» в компании Xerox, а за технологией Token Ring стояла компания IBM. Первые стандарты технологий локальных сетей также были фирменными, что было, естественно, не очень удобно как для пользователей, так и для компаний-производителей сетевого оборудования. Для исправления ситуации в 1980 году в институте IEEE был организован комитет 802 по стандартизации технологий LAN. Результатом работы комитета IEEE 802 стало принятие семейства стандартов IEEE 802.x, содержащих

рекомендации по проектированию нижних уровней локальных сетей. Эти стандарты базировались на обобщении популярных фирменных стандартов, в частности, Ethernet и Token Ring.

Комитет IEEE 802 и сегодня является основным международным органом, разрабатывающим стандарты технологий локальных сетей, в том числе коммутируемых локальных сетей, а также стандарты беспроводных локальных сетей на разделяемой среде. Помимо IEEE в работе по стандартизации протоколов LAN принимали и принимают участие и другие организации. Так, для сетей, работающих на оптоволокне, институтом ANSI был разработан стандарт FDDI, обеспечивающий скорость передачи данных 100 Мбит/с. Это был первый протокол LAN, который достиг такой скорости, в 10 раз превысив скорость технологии Ethernet.

Структуру стандартов IEEE 802 иллюстрирует рис. 6.1.

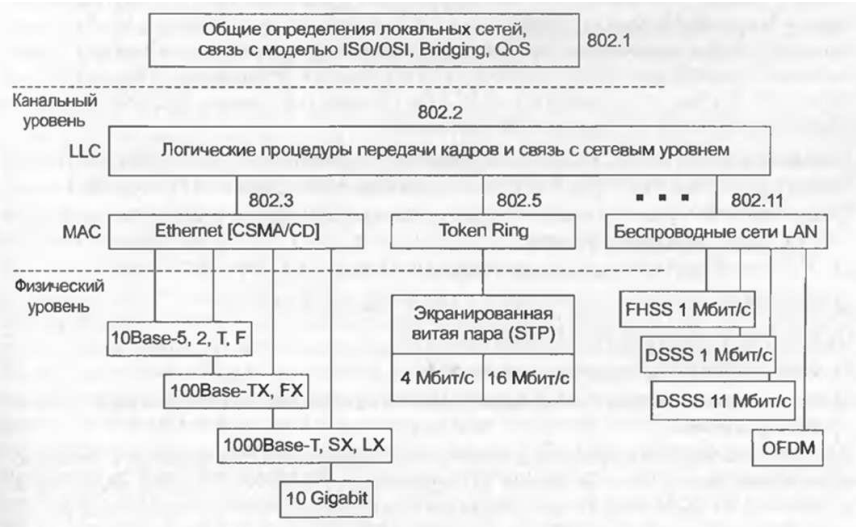


Рис. 6.1 — Структура стандартов IEEE 802.x

Стандарты IEEE 802 описывают функции, которые можно отнести к функциям физического и канального уровней модели OSI. Как видно из рисунка 6.1, эти стандарты имеют как общие для всех технологий части, так и индивидуальные.

Общую группу стандартов составляют стандарты рабочей группы 802.1. Эти стандарты описывают наиболее высокоуровневые функции локальных сетей. Так, в документах 802.1 даются общие определения локальных сетей и их свойств, показана связь трех уровней модели IEEE 802 с моделью OSI. Наиболее практически важными являются те стандарты рабочей группы 802.1, которые описывают взаимодействие различных технологий, а также стандарты по построению более сложных сетей на основе базовых топологий. Эта группа стандартов носит общее название стандартов межсетевого взаимодействия. Наиболее важным в настоящее время является стандарт 802.1 D, описывающий логику работы прозрачного моста, которая лежит в основе любого современного коммутатора Ethernet (и лежала бы в основе коммутатора Token Ring или FDDI, если бы они сохранились до наших дней). Набор стандартов, разработанных рабочей группой 802.1, продолжает расти, в настоящее время это наиболее активный подкомитет комитета 802. Например, этот комитет стандартизовал технологию виртуальных локальных сетей, также он занимается стандартизацией технологий, известных под общим названием Carrier Ethernet.

Каждая из рабочих групп 802.3, 802.4, 802.5 и т. д. ответственна за стандартизацию конкретной технологии, например группа 802.3 занимается технологией Ethernet, группа 802.4 — технологией ArcNet, группа 802.5 — технологией Token Ring, группа 802.11 — технологией беспроводных локальных сетей. Стандарты этих рабочих

групп описывают как физический уровень (или несколько возможных физических уровней), так и канальный уровень конкретной технологии (последний включает описание метода доступа, используемого технологией). Основу стандарта 802.3 составила технология экспериментальной сети Ethernet Network, которую фирма Xerox разработала и реализовала в 1975 году. В 1980 году фирмы DEC, Intel и Xerox (сокращенно — DIX) совместно разработали и опубликовали стандарт Ethernet версии II для сети, построенной на основе коаксиального кабеля. Эту последнюю версию фирменного стандарта Ethernet называют стандартом Ethernet DIX, или Ethernet II. На базе стандарта Ethernet DIX был разработан стандарт IEEE 802.3, который во многом совпадает со своим предшественником.

Помимо индивидуальных для каждой технологии уровней существует и общий уровень, который был стандартизован рабочей группой 802.2. Появление этого уровня связано с тем, что комитет 802 разделил функции канального уровня модели OSI на два уровня:

- управление логическим каналом (Logical Link Control, LLC);
- управление доступом к среде (Media Access Control, MAC). Основными функциями уровня MAC являются:
- обеспечение доступа к разделяемой среде;
- передача кадров между конечными узлами посредством функций и устройств физического уровня.

Если уровень MAC специфичен для каждой технологии и отражает различия в методах доступа к разделяемой среде, то уровень LLC представляет собой обобщение функций разных технологий по обеспечению передачи кадра с различными требованиями к надежности.

Логика образования общего для всех технологий уровня LLC заключается в следующем: после того как узел сети получил доступ к среде в соответствии с алгоритмом, специфическим для конкретной технологии, дальнейшие действия узла или узлов по обеспечению надежной передачи кадров от технологии не зависят.

Так как в зависимости от требований приложения может понадобиться разная степень надежности, то рабочая группа 802.2 определила три типа услуг

- Услуга LLC1 — это услуга без установления соединения и без подтверждения получения данных. LLC1 дает пользователю средства для передачи данных с минимумом издержек. В этом случае LLC поддерживает дейтаграммный режим работы, как и MAC, так что и технология LAN в целом работает в дейтаграммном режиме. Обычно эта процедура используется, когда такие функции, как восстановление данных после ошибок и упорядочивание данных, выполняются протоколами вышележащих уровней, поэтому нет нужды дублировать их на уровне LLC.
- Услуга LLC2 дает пользователю возможность установить логическое соединение перед началом передачи любого блока данных и, если это требуется, выполнить процедуры восстановления после ошибок и упорядочивание потока блоков в рамках установленного соединения.
- Услуга LLC3 — это услуга без установления соединения, но с подтверждением получения данных. В некоторых случаях (например, при использовании сетей в системах реального времени, управляющих промышленными объектами), с одной стороны, временные издержки установления логического соединения

перед отправкой данных неприемлемы, а с другой стороны, подтверждение о корректности приема переданных данных необходимо. Для такого рода ситуаций и предусмотрена дополнительная услуга LLC3, которая является компромиссом между LLC1 и LLC2, так как она не предусматривает установление логическою соединения, но обеспечивает подтверждение получения данных. Какой из трех режимов работы уровня LLC будет использован, зависит от требований протокола верхнего уровня. Информация о требуемой от LLC транспортной услуге передается через межуровневый интерфейс уровню LLC вместе с аппаратным адресом и пакетом с пользовательскими данными. Например, когда поверх LLC работает протокол IP, он всегда запрашивает режим LLC1, поскольку в стеке TCP/IP задачу обеспечения надежной доставки решает протокол TCP

Нужно сказать, что на практике идея обобщения функций обеспечения надежной передачи кадров в общем уровне LLC не оправдала себя. Технология Ethernet в версии DIX изначально функционировала в наиболее простом дейтаграммном режиме — в результате оборудование Ethernet и после опубликования стандарта IEEE 802.2 продолжало поддерживать только этот режим работы, который формально является режимом LLC1. В то же время оборудование сетей Token Ring, которое изначально поддерживало режимы LLC2 и LLC3, также продолжало поддерживать эти режимы и никогда не поддерживало режим LLC1. Помимо обеспечения заданной степени надежности уровень LLC выполняет также интерфейсные функции. Эти функции заключаются в передаче пользовательских и служебных данных между

уровнем MAC и сетевым уровнем. При передаче данных сверху вниз уровень LLC принимает от протокола сетевого уровня пакет (например, IP- или IPX-пакет), в котором уже находятся пользовательские данные. Помимо пакета сверху также передается адрес узла назначения в формате той технологии LAN, которая будет использована для доставки кадра в пределах данной локальной сети. Напомним, что в терминах стека TCP/IP такой адрес называется аппаратным. Полученные от сетевого уровня пакет и аппаратный адрес уровень LLC передает далее вниз — уровню MAC. Кроме того, LLC при необходимости решает задачу мультиплексирования, передавая данные от нескольких протоколов сетевого уровня единственному протоколу уровня MAC.

При передаче данных снизу вверх LLC принимает от уровня MAC пакет сетевого уровня, пришедший из сети. Теперь ему нужно выполнить еще одну интерфейсную функцию — демультиплексирование, то есть решить, какому из сетевых протоколов передать полученные от MAC данные.

MAC-адреса

На уровне MAC, который обеспечивает доступ к среде и передачу кадра, для идентификации сетевых интерфейсов узлов сети используются регламентированные стандартом IEEE 802.3 уникальные 6-байтовые адреса, называемые MAC-адресами. Обычно MAC-адрес записывают в виде шести пар шестнадцатеричных цифр, разделенных тире или двоеточиями, например 11-A0-17-3D-BC-01. Каждый сетевой адаптер имеет, по крайней мере, один MAC-адрес.

Помимо отдельных интерфейсов, MAC-адрес может определять группу интерфейсов или даже все интерфейсы сети. Первый (младший) бит старшего байта адреса назначения является признаком того, является адрес

индивидуальным или групповым. Если он равен 0, то адрес является индивидуальным, то есть идентифицирует один сетевой интерфейс, а если 1, то групповым. Групповой адрес связан только с интерфейсами, сконфигурированными (вручную или автоматически но запросу вышележащего уровня) как члены группы, номер которой указан в групповом адресе. Если сетевой интерфейс включен в группу, то наряду с уникальным MAC-адресом с ним ассоциируется еще один адрес — групповой. В частном случае, если групповой адрес состоит из всех единиц, он идентифицирует все узлы сети и называется широковещательным.

Второй бит старшего байта адреса определяет способ назначения адреса — централизованный или локальный. Если этот бит равен 0 (что бывает почти всегда в стандартной аппаратуре Ethernet), это говорит о том, что адрес назначен централизованно по правилам IEEE 802.

В стандартах IEEE Ethernet младший бит байта изображается в самой левой позиции поля, а старший бит — в самой правой. Этот нестандартный способ отображения порядка следования битов в банте соответствует порядку передачи битов в линию связи передатчиком Ethernet (первым перелается младший бит). В стандартах других организаций, например RFC, IETF, ITU-T, ISO, используется традиционное представление байта, когда младший бит считается самым правым битом байта, а старший самым левым. При этом порядок следования байтов остается традиционным. Поэтому при чтении стандартов, опубликованных этими организациями, а также чтении данных, отображаемых на экране операционной системой или анализатором протоколов, значения каждого байта кадра Ethernet нужно зеркально отобразить, чтобы получить представление о значении разрядов того байта в соответствии с документами IEEE.

Например, групповой адрес, имеющий в нотации IEEE вид 1000 0000 0000 0000 1010 0111 1111 0000 0000 0000 0000 0000 или в шестнадпатеричной записи 80-00-A7-F0-00-00.

Комитет IEEE распределяет между производителями оборудования так называемые организационно уникальные идентификаторы (Organizationally Unique Identifier, OUI). Каждый производитель помещает выделенный ему идентификатор в три старших байта адреса. За уникальность младших трех байтов адреса отвечает производитель оборудования. Двадцать четыре бита, отводимые производителю для адресации интерфейсов его продукции, позволяют выпустить примерно 16 миллионов интерфейсов под одним идентификатором организации. Уникальность централизованно распределяемых адресов распространяется на все основные технологии локальных сетей — Ethernet, Token Ring, FDDI и т.д. Локальные адреса назначаются администратором сети, в обязанности которого входит обеспечение их уникальности.

Сетевые адаптеры Ethernet могут также работать в так называемом режиме неразборчивого захвата (promiscuous mode), когда они захватывают все кадры, поступающие на интерфейс, независимо от их MAC-адресов назначения. Обычно такой режим используется для мониторинга трафика, когда захваченные кадры изучаются затем для нахождения причины некорректного поведения некоторого узла или отладки нового протокола.

Существует несколько стандартов формата кадра Ethernet. На практике в оборудовании Ethernet используется только один формат кадра, а именно кадр Ethernet DIX, который иногда называют кадром Ethernet II по номеру последнего стандарта DIX.

Для доставки кадра достаточно одного адреса — адреса назначения; адрес источника помещается в кадр для

того, чтобы узел, получивший кадр, знал, от кого пришел кадр и кому нужно на него ответить. Принятие решения об ответе не входит в компетенцию протокола Ethernet, это дело протоколов верхних уровней. Ethernet же только выполнит такое действие, если с сетевого уровня поступит соответствующее указание.

Доступ к среде и передача данных

Метод доступа, используемый в сетях Ethernet на разделяемой проводной среде, носит название CSMA/CD (Carrier Sense Multiple Access with Collision Detection — прослушивание несущей частоты с множественным доступом и распознаванием коллизий). Название метода достаточно хорошо описывает его особенности.

Все компьютеры в сети на разделяемой среде имеют возможность немедленно (с учетом задержки распространения сигнала в физической среде) получить данные, которые любой из компьютеров начал передавать в общую среду. Говорят, что среда, к которой подключены все станции, работает в режиме коллективного доступа (Multiply Access, MA). Чтобы получить возможность передавать кадр, интерфейс-отправитель должен убедиться, что разделяемая среда свободна. Это достигается прослушиванием основной гармоники сигнала, которая еще называется несущей частотой (Carrier Sense, CS).

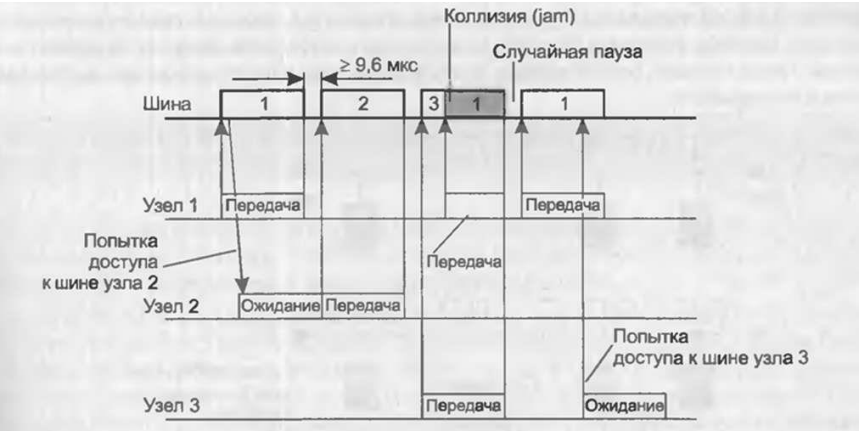


Рис. 6.2 — Метод случайного доступа CSMA/CD.

Признаком «незанятости» среды является отсутствие на ней несущей частоты, которая при манчестерском способе кодирования равна 5-10 МГц в зависимости от последовательности единиц и нулей, передаваемых в данный момент.

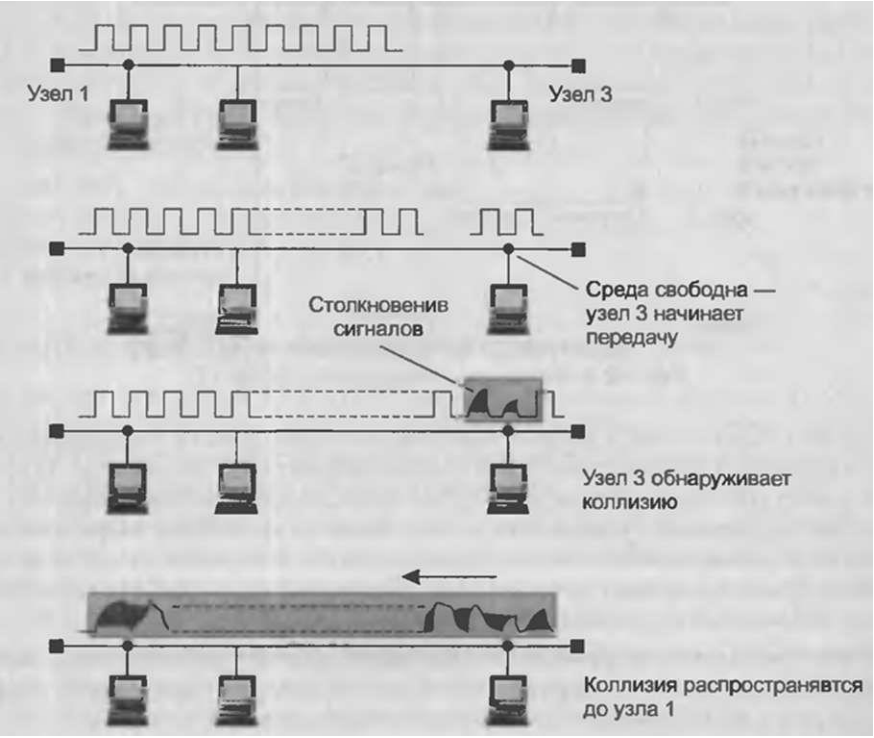


Рис. 6.3 — Схема возникновения и распространения коллизии

Если среда свободна, то узел имеет право начать передачу кадра. В примере, показанном на рис. 6.2, узел 1 обнаружил, что среда свободна, и начал передавать свой кадр. В коаксиальном кабеле сигналы передатчика узла 1 распространяются в обе стороны, так что их получают все узлы сети. Кадр данных всегда сопровождается преамбулой, которая состоит из 7 байт, каждый из которых имеет значение 10101010, и 8-го байта, равного 10101011. Последний байт носит название ограничителя начала кадра. Преамбула нужна для вхождения приемника в побитовую и побайтовую синхронизацию с передатчиком. Наличие двух единиц, идущих подряд, говорит приемнику

о том, что преамбула закончилась и следующий бит является началом кадра.

Все станции, подключенные к кабелю, начинают записывать байты передаваемого кадра в свои внутренние буферы. Первые 6 байт кадра содержат адрес назначения. Та станция, которая узнает собственный адрес в заголовке кадра, продолжает записывать его содержимое в свой внутренний буфер, а остальные станции на этом прием кадра прекращают. Станция назначения обрабатывает полученные данные и передает их вверх по своему стеку. Кадр Ethernet содержит не только адрес назначения, но и адрес источника данных, поэтому станция-получатель знает, кому нужно послать ответ.

Узел 2 во время передачи кадра узлом 1 также пытался начать передачу своего кадра, однако обнаруживает, что среда занята — на ней присутствует несущая частота, — поэтому узел 2 вынужден ждать, пока узел 1 не прекратит передачу кадра.

После окончания передачи кадра все узлы сети обязаны выдержать технологическую паузу, равную межпакетному интервалу (Inter Packet Gap, IPG) в 9,6 мкс. Эта пауза нужна для приведения сетевых адаптеров в исходное состояние, а также для предотвращения монопольного захвата среды одной станцией. После окончания технологической паузы узлы имеют право начать передачу своего кадра, так как среда свободна. В приведенном примере узел 2 дождался окончания передачи кадра узлом 1, сделал паузу в 9,6 мкс и начал передачу своего кадра

Возникновение коллизии

Механизм прослушивания среды и пауза между кадрами не гарантируют исключения ситуации, когда две или более станции одновременно решают, что среда свободна, и начинают передавать свои кадры. Говорят, что

при этом происходит коллизия, так как содержимое обоих кадров сталкивается на общем кабеле и происходит искажение информации.

Коллизия — это нормальная ситуация в работе сетей Ethernet. В примере на рис. 6.3 коллизию породила одновременная передача данных узлами 3 и 1. Для возникновения коллизии не обязательно, чтобы несколько станций начали передачу абсолютно одновременно, такая ситуация маловероятна. Более вероятна ситуация, когда один узел начинает передачу, а через некоторое (короткое) время другой узел, проверив среду и не обнаружив несущую (сигналы первого узла еще не успели до него дойти), начинает передачу своего кадра. Таким образом, возникновение коллизии является следствием распределения узлов сети в пространстве.

Чтобы корректно обработать коллизию, все станции одновременно наблюдают за возникающими на кабеле сигналами. Если передаваемые и наблюдаемые сигналы отличаются, то фиксируется факт обнаружения коллизии (Collision Detection, CD). Для повышения вероятности скорейшего обнаружения коллизии всеми станциями сети станция, которая обнаружила коллизию, прерывает передачу своего кадра (в произвольном месте, возможно, и не на границе байта) и усугубляет коллизию посылкой в сеть специальной последовательности из 32 бит, называемой jam-последовательностью.

После этого обнаружившая коллизию передающая станция обязана прекратить передачу и сделать паузу в течение короткого случайного интервала времени. Затем она может снова предпринять попытку захвата среды и передачи кадра. Случайная пауза выбирается по следующему алгоритму:

$$\text{Пауза} = L \times (\text{интервал отсрочки})$$

В технологии Ethernet интервал отсрочки равен значению 512 битовых интервалов. Битовый интервал соответствует времени между появлением двух последовательных битов данных на кабеле; для скорости 10 Мбит/с величина битового интервала равна 0,1 мкс, или 100 нс.

L — представляет собой целое число, выбранное с равной вероятностью из диапазона [0,2N), гдеN — номер повторной попытки передачи данного кадра: 1,2,..., 10. После 10-й попытки интервал, из которого выбирается пауза, не увеличивается.

Таким образом, случайная пауза в технологии Ethernet может принимать значения от 0 до 52,4 мс.

Если 16 последовательных попыток передачи кадра вызывают коллизию, то передатчик должен прекратить попытки и отбросить этот кадр. Описанный алгоритм носит название усеченного экспоненциального двоичного алгоритма отсрочки.

Администраторы сетей Ethernet на разделяемой среде руководствуются простым эмпирическим правилом — коэффициент использования среды не должен превышать 30 %. Для поддержки чувствительного к задержкам трафика сети Ethernet (и другие сети на разделяемой среде) могут применять только один метод поддержания характеристик QoS — недогруженный режим работы.

Время оборота и распознавание коллизий

Надежное распознавание коллизий всеми станциями сети является необходимым условием корректной работы сети Ethernet. Если какая-либо передающая станция не распознает коллизию и решит, что кадр данных передан ею верно, этот кадр будет утерян. Из-за наложения сигналов при коллизии информация кадра исказится, и он будет отбракован принимающей станцией

из-за несовпадения контрольной суммы. Скорее всего, недошедшие до получателя данные будут повторно переданы каким-либо протоколом верхнего уровня, например транспортным или прикладным, работающим с установлением соединения, либо протоколом LLC, если он работает в режиме LLC2. Однако повторная передача сообщения протоколами верхних уровней произойдет гораздо позже (иногда по прошествии нескольких секунд), чем повторная передача средствами сети Ethernet, работающей с микросекундными интервалами. Поэтому если коллизии не будут надежно распознаваться узлами сети Ethernet, то это приведет к заметному снижению полезной пропускной способности сети.

Для надежного распознавания коллизий должно выполняться следующее соотношение:

$$T_{min} \geq RTT$$

Здесь T_{min} — время передачи кадра минимальной длины, а RTT — время оборота, то есть время, за которое сигнал коллизии успевает распространиться до самого дальнего узла сети. В худшем случае сигнал должен пройти дважды между наиболее удаленными друг от друга станциями сети (в одну сторону проходит неискаженный сигнал, а в обратном направлении — сигнал, уже искаженный коллизией).

При выполнении этого условия передающая станция должна успеть обнаружить коллизию, которую вызвал переданный ее кадр, еще до того, как она закончит передачу этого кадра. Очевидно, что выполнение этого условия зависит, с одной стороны, от минимальной длины кадра и скорости передачи данных протокола, а с другой стороны, от длины кабельной системы сети и скорости распространения сигнала в кабеле (для разных типов кабеля эта скорость несколько отличается).

Все параметры протокола Ethernet подобраны таким образом, чтобы при нормальной работе коллизии четко распознавались. Так, стандарт Ethernet определяет минимальную длину поля данных кадра в 46 байт (что вместе со служебными полями дает минимальную длину кадра 64 байт, а вместе с преамбулой — 72 байт, или 576 бит). Отсюда может быть вычислено ограничение на расстояние между станциями. В стандарте Ethernet 10 Мбит/с время передачи кадра минимальной длины равно 575 битовых интервалов, следовательно, время оборота должно быть меньше 57,5 мкс. Расстояние, которое сигнал может пройти за это время, зависит от типа кабеля и для толстого коаксиального кабеля равно примерно 13 280 м. Учитывая, что за время 57,5 мкс сигнал должен пройти по линии связи дважды, расстояние между двумя узлами не должно быть больше 6635 м. В стандарте величина этого расстояния выбрана равной 2500 м, что существенно меньше. Это объясняется тем, что повторители, которые нужны для соединения пяти сегментов кабеля, вносят задержки в распространение сигнала.

Описанные соображения объясняют выбор минимальной длины поля данных кадра в 46 байт. Уменьшение этого значения до 0 привело бы к значительному сокращению максимальной длины сети.

Требование $T_{min} \geq RTT$ имеет одно интересное следствие: чем выше скорость протокола, тем меньше должна быть максимальная длина сети. Поэтому для Ethernet на разделяемой среде при скорости в 100 Мбит/с максимальная длина сети пропорционально уменьшается до 250 м, а при скорости в 1 Гбит/с — до 25 м. Эта зависимость, наряду с резким ростом задержек при повышении загрузки сети, говорит о еще одном коренном недостатке метода доступа CSMA/CD.

Спецификации физической среды

При стандартизации технологии Ethernet рабочей группой IEEE 802.3 вариант Ethernet на «толстом» коаксиальном кабеле получил название 10Base-5.

Число 10 в этом названии обозначает номинальную битовую скорость передачи данных стандарта, то есть 10 Мбит/с, а слово «Base» — метод передачи на одной базовой частоте (в данном случае 10 МГц). Последний символ в названии стандарта физического уровня обозначает тип кабеля, в данном случае 5 отражает тот факт, что диаметр «толстого» коаксиала равен 0,5 дюйма. Данная система обозначения тина физического уровня Ethernet сохранилась до настоящего времени.

Наиболее популярными спецификациями физической среды Ethernet для скорости передачи данных 10 Мбит/с являются следующие:

- 10Base-5 — коаксиальный кабель диаметром 0,5 дюйма, называемый «толстым» коаксиалом. Имеет волновое сопротивление 50 Ом. Максимальная длина сегмента: 500 м (без повторителей). Максимальное количество узлов подключаемых к сегменту — 100. Максимальное число сегментов — 5 (4 повторителя), из которых только 3 могут использоваться для подключения узлов, а 2 играют роль удлинителей сети.
- 10Base-2 — коаксиальный кабель диаметром 0,25 дюйма, называемый «тонким» коаксиалом. Имеет волновое сопротивление 50 Ом. Максимальная длина сегмента — 185 м (без повторителей). Максимальное количество узлов подключаемых к сегменту — 30 Максимальное число сегментов — 5 (4 повторителя), из которых только 3 могут использоваться для подключения узлов, а 2 играют роль удлинителей сети.

- 10Base-T — кабель на основе неэкранированной витой пары (UTP). Образует звездообразную топологию на основе концентратора (многопортового повторителя). Расстояние между концентратором и конечным узлом — не более 100 м. Между любыми двумя узлами сети может быть не более 4-х концентраторов (так называемое «правило 4-х хабов»).
- 10Base-F — волоконно-оптический кабель. Топология аналогична топологии стандарта 10Base-T, но расстояние между концентратором и конечным узлом может достигать 2000 м. Правило 4-х хабов остается в силе. В стандарте 10Base-2 в качестве передающей среды используется «тонкий» коаксиал Ethernet. Станции подключаются к кабелю с помощью высокочастотного Т-коннектора, представляющего собой тройник, один отвод которого соединяется с сетевым адаптером, а два других — с двумя концами разрыва кабеля. Стандарт 10Base-2 очень близок к стандарту 10Base-5, но трансиверы в нем объединены с сетевыми адаптерами за счет того, что более гибкий тонкий коаксиальный кабель может быть подведен непосредственно к выходному разъему платы сетевого адаптера, установленной в шасси компьютера. Кабель в данном случае «висит» на сетевом адаптере, что затрудняет физическое перемещение компьютеров, однако сама операция соединения компьютеров в сеть оказывается гораздо проще, чем для сети на «толстом» коаксиале.

Реализация этого стандарта на практике приводит к наиболее простому решению для кабельной сети, так как

для соединения компьютеров требуются только сетевые адаптеры, Т-коннекторы и терминаторы на 50 Ом. Однако этот вид кабельных соединений наиболее сильно подвержен авариям и сбоям. Кабель более восприимчив к помехам, чем «толстый» коаксиал. В моноканале имеется большое количество механических соединений: каждый Т-коннектор дает три механических соединения, два из которых имеют жизненное значение для всей сети. Пользователи имеют доступ к разъемам и могут нарушить целостность моноканала. Кроме того, эстетика и эргономичность этого решения оставляют желать лучшего, так как от каждой станции через Т-коннектор отходят два довольно заметных провода, которые под столом часто образуют моток кабеля — запас, необходимый на случай даже небольшого перемещения рабочего места.

Сеть Eternet па витой паре, описываемая стандартом 10Base-T, стала следующим шагом на пути повышения эксплуатационных характеристик Ethernet.

Одним из существенных недостатков Ethernet на коаксиальном кабеле являлось отсутствие оперативной информации о состоянии кабеля и сложность нахождения места его повреждения. Поэтому поиск неисправностей стал привычной процедурой и головной болью многочисленной армии сетевых администраторов коаксиальных сетей Ethernet. Альтернатива появилась в середине 80-х годов, когда благодаря использованию витой пары и повторителей сети Ethernet стали гораздо более ремонтопригодными.

К этому времени телефонные компании уже достаточно давно применяли многопарный кабель на основе неэкранированной витой пары для подключения телефонных аппаратов внутри зданий. Идея приспособить этот популярный вид кабеля для локальных сетей оказалась очень плодотворной, так как многие здания уже

были оснащены нужной кабельной системой. Оставалось разработать способ подключения сетевых адаптеров и прочего коммуникационного оборудования к витой паре таким образом, чтобы изменения в сетевых адаптерах и программном обеспечении сетевых операционных систем были минимальными по сравнению с сетями Ethernet на коаксиале. Эта попытка оказалась успешной — переход на витую пару требует только замены приемника и передатчика сетевого адаптера, а метод доступа и все протоколы канального уровня остаются теми же, что и в сетях Ethernet на коаксиале.

Правда, для соединения узлов в сеть теперь обязательно требуется коммуникационное устройство — многопортовый повторитель Ethernet на витой паре.

Многопортовый повторитель часто называют концентратором, или хабом (от английского hub — центр, ступица колеса), так как в нем сконцентрированы соединения со всеми конечными узлами сети. Фактически хаб имитирует сеть на коаксиальном кабеле в том отношении, что физически отдельные отрезки кабеля на витой паре логически все равно представляют единую разделяемую среду. Все правила доступа к среде по алгоритму CSMA/CD сохраняются.

При создании сети Ethernet на витой паре с большим числом конечных узлов хабы можно соединять друг с другом иерархическим способом, образуя древовидную структуру. Добавление каждого хаба изменяет физическую структуру, но оставляет без изменения логическую структуру сети. То есть независимо от числа хабов в сети сохраняется одна общая для всех интерфейсов разделяемая среда, так что передача кадра с любого интерфейса блокирует передатчики всех остальных интерфейсов.

Физическая структуризация сетей, построенных на основе витой пары, повышает надежность и упрощает обслуживание сети, поскольку в этом случае появляется возможность контролировать состояние и локализовывать отказы отдельных кабельных отрезков, подключающих конечные узлы к концентраторам. В случае обрыва, короткого замыкания или неисправности сетевого адаптера работа сети может быть быстро восстановлена путем отключения соответствующего сегмента кабеля.

Для контроля целостности физического соединения между двумя непосредственно соединенными портами в стандарте 10Base-T введен так называемый тест целостности соединения (Link Integrity Test, LIT). Эта процедура заключается в том, что в те периоды, когда порт не посылает или получает кадры данных, он посылает своему соседу импульсы длительностью 100 нс через каждые 16 мс. Если порт принимает такие импульсы от своего соседа, то он считает соединение работоспособным и, как правило, индицирует это зеленым светом светодиода.

Незааисимо от используемого физического уровня в стандартах Ethernet на 10 Мбит/с вводится ограничение на максимальное количество узлов, подключаемых к разделяемой среде. Это ограничение составляет 1024 узла.

Не все варианты физического уровня стандарта Ethernet на 10 Мбит/с дают возможность построить сеть с максимальным количеством узлов. Например, сеть 10Base-5 может иметь максимум 100 х 3 - 3 - 296 узлов (4 подключения уходят на повторители, соединяющие сегменты), а сеть 10 Base-2 — только 87 узлов. И лишь сети 10Base-T и 10Base-F дают такую возможность.

Максимальная производительность сети Ethernet

Производительность сети зависит от скорости передачи кадров по линиям связи и скорости обработки этих кадров коммуникационными устройствами, передающими кадры между своими портами, к которым эти линии связи подключены. Скорость передачи кадров по линиям связи зависит от используемых протоколов физического и канального уровней, например Ethernet на 10 Мбит/с, Ethernet на 100 Мбит/с, Token Ring или FDDI.

Скорость, с которой протокол передает биты по линии связи, называется номинальной скоростью протокола.

Скорость обработки кадров коммуникационным устройством зависит от производительности его процессоров, внутренней архитектуры и других параметров. Очевидно, что скорость коммуникационного устройства должна соответствовать скорости работы линии. Если она меньше скорости работы линии, то кадры будут стоять в очередях и отбрасываться при переполнении последних. В то же время нет смысла применять устройство, которое в сотни раз производительнее, чем того требует скорость подключаемых к нему линий.

Для оценки требуемой производительности коммуникационных устройств, имеющих порты Ethernet, необходимо оценить производительность сегмента Ethernet, но не в битах в секунду (ее мы знаем — это 10 Мбит/с), а в кадрах в секунду, так как именно этот показатель помогает оценить требования к производительности коммуникационных устройств. Это объясняется тем, что на обработку каждого кадра, независимо от его длины, мост, коммутатор или маршрутизатор тратит примерно равное время, которое уходит на просмотр таблицы продвижения пакета, формирование нового кадра (для маршрутизатора) и т.п.

При постоянной битовой скорости количество кадров, поступающих на коммуникационное устройство в единицу времени, является, естественно, максимальным при их минимальной длине. Поэтому для коммуникационного оборудования наиболее тяжелым режимом является обработка потока кадров минимальной длины.

Технологии Token Ring и FDDI

Token Ring и FDDI — это функционально намного более сложные технологии, чем Ethernet на разделяемой среде. Разработчики этих технологий стремились наделить сеть на разделяемой среде многими положительными качествами: сделать механизм разделения среды предсказуемым и управляемым, обеспечить отказоустойчивость сети, организовать приоритетное обслуживание для чувствительного к задержкам трафика, например голосового. Нужно отдать им должное — во многом их усилия оправдались, и сети FDDI довольно долгое время успешно использовались как магистрали сетей масштаба кампуса, в особенности в тех случаях, когда нужно было обеспечить высокую надежность магистрали.

Механизм доступа к среде в сетях Token Ring и FDDI является более детерминированным, чем в сетях Ethernet.

Рассмотрим его на примере сети Token Ring, станции которой связаны в кольцо, так что любая станция непосредственно получает данные только от одной станции — той, которая является предыдущей в кольце, а передает данные своему ближайшему соседу вниз по потоку данных. Скорость передачи данных в первых сетях Token Ring, разработанных компанией IBM, была всего 4 Мбит/с, но затем была повышена до 16 Мбит/с. Основная среда передачи данных — витая пара. Для адресации

станций сети Token Ring (и FDDI) используют MAC-адреса того же формата, что и Ethernet.

Метод доступа Token Ring основан на передаче от узла к узлу специального кадра — токена, или маркера, доступа, при этом только узел, владеющий токеном, может передавать свои кадры в кольцо, которое становится в этом случае разделяемой средой. Существует лимит на период монопольного использования среды — это так называемое время удержания токена, по истечение которого станция обязана передать токен своему соседу по кольцу. В результате такие ситуации, как неопределенное время ожидания доступа к среде, характерные для Ethernet, здесь исключены (по крайней мере, в тех случаях, когда сетевые адаптеры станций исправны и работают без сбоев). Максимальное время ожидания всегда нетрудно оценить, так как оно равно произведению времени удержания токена на количество станций в кольце. Так как станция, получившая токен, но не имеющая в этот момент кадров для передачи, передает токен следующей станции, то время ожидания может быть меньше.

Отказоустойчивость сети Token Ring определяется использованием в сети повторителей для создания кольца. Каждый такой повторитель имеет несколько портов, которые образуют кольцо за счет внутренних связей между передатчиками и приемниками. В случае отказа или отсоединения станции повторитель организует обход порта этой станции, так что связность кольца не нарушается.

Поддержка чувствительного к задержкам трафика достигается за счет системы приоритетов кадров. Решение о приоритете конкретного кадра принимает передающая станция. Токен также всегда имеет некоторый уровень текущего приоритета. Станция имеет право захватить переданный ей токен только в том случае, если приоритет

кадра, который она хочет передать, выше приоритета токена (или равен ему). В противном случае станция обязана передать токен следующей по кольцу станции.

Благодаря более высокой, чем в сетях Ethernet, скорости, детерминированности распределения пропускной способности сети между узлами, а также лучших эксплуатационных характеристик (обнаружение и изоляция неисправностей), сети Token Ring были предпочтительным выбором для таких чувствительных к подобным показателям приложений, как банковские системы и системы управления предприятием.

Технологию FDDI можно считать усовершенствованным вариантом Token Ring, так как в ней, как и в Token Ring, используется метод доступа к среде, основанный на передаче токена, а также кольцевая топология связей, но вместе с тем FDDI работает на более высокой скорости и имеет более совершенный механизм отказоустойчивости. Технология FDDI стала первой технологией локальных сетей, в которой оптическое волокно, начавшее применяться в телекоммуникационных сетях с 70-х годов прошлого века, было использовано в качестве разделяемой среды передачи данных. За счет применения оптических систем скорость передачи данных удалось повысить до 100 Мбит/с (позже появилось оборудование FDDI на витой паре, работающее на той же скорости). В тех случаях, когда нужно было обеспечить высокую надежность сети FDDI, применялось двойное кольцо. В нормальном режиме станции используют для передачи данных и токена доступа первичное кольцо, а вторичное простаивает. В случае отказа, например, при обрыве кабеля между станциями, первичное кольцо объединяется со вторичным, вновь образуя единое кольцо. Этот режим работы сети называется режимом свертывания колец. Операция свертывания производится средствами

повторителей (не показанных на рисунке) и/или сетевых адаптеров FDDI. Для упрощения этой процедуры данные по первичному кольцу всегда передаются в одном направлении (на диаграммах это направление изображается против часовой стрелки), а по вторичному — в обратном (изображается по часовой стрелке). Поэтому при образовании общего кольца из двух колец передатчики станций по-прежнему остаются подключенными к приемникам соседних станций, что позволяет правильно передавать и принимать информацию соседними станциями.

В стандартах FDDI много внимания отводится различным процедурам, которые позволяют определить факт наличия отказа в сети, а затем произвести необходимое реконфигурирование. Технология FDDI расширяет механизмы обнаружения отказов технологии Token Ring за счет резервных связей, которые предоставляет второе кольцо.

Лекция 7. Адресация в стеке протоколов TCP/IP.

Важную часть технологии TCP/IP составляют задачи адресации, к числу которых относятся следующие:

- Согласованное использование адресов различного типа. Эта задача включает отображение адресов разных типов, например преобразование сетевого IP-адреса в локальный, доменного имени — в IP-адрес.
- Обеспечение уникальности адресов. В зависимости от типа адреса требуется обеспечивать однозначность адресации в пределах компьютера, подсети, корпоративной сети или Интернета.
- Конфигурирование сетевых интерфейсов и сетевых приложений.

Каждая из перечисленных задач имеет достаточно простое решение для сети, число узлов которой не превосходит нескольких десятков. Например, для отображения символьного доменного имени на IP-адрес достаточно поддерживать на каждом хосте таблицу всех символьных имен, используемых в сети, и соответствующих им IP-адресов. Столь же просто «вручную» присвоить всем интерфейсам в небольшой сети уникальные адреса. Однако в крупных сетях эти же задачи усложняются настолько, что требуют принципиально других решений.

Ключевым словом, которое характеризует подход к решению этих проблем, принятый в TCP/IP, является масштабируемость.

Процедуры, предлагаемые TCP/IP для назначения, отображения и конфигурирования адресов, одинаково хорошо работают в сетях разного масштаба. В этой главе

наряду с собственно схемой образования IP-адресов мы познакомимся с наиболее популярными масштабируемыми средствами поддержки адресации в сетях TCP/IP: технологией бесклассовой междоменной маршрутизации, системой доменных имен, протоколом динамического конфигурирования хостов.

Сегодня стек TCP/IP широко используется как в глобальных, так и в локальных сетях. Этот стек имеет иерархическую структуру, в которой определено 4 уровня.

Прикладной уровень стека TCP/IP соответствует трем верхним уровням модели OSI: прикладному, представления и сеансовому. Он объединяет сервисы, предоставляемые системой пользовательским приложениям. За долгие годы применения в сетях различных стран и организаций стек TCP/IP накопил большое количество протоколов и служб прикладного уровня. К ним относятся такие распространенные протоколы, как протокол передачи файлов (File Transfer Protocol, FTP), протокол эмуляции терминала telnet, простой протокол передачи почты (Simple Mail Transfer Protocol, SMTP), протокол передачи гипертекста (Hypertext Transfer Protocol, HTTP) и многие другие. Протоколы прикладного уровня развертываются на хостах.

Транспортный уровень стека TCP/IP может предоставлять вышележащему уровню два типа сервиса:

- гарантированную доставку обеспечивает протокол управления передачей (Transmission Control Protocol, TCP);
- доставку по возможности, или с максимальными усилиями, обеспечивает протокол пользовательских дейтаграмм (User Datagram Protocol, UDP).

Для того чтобы обеспечить надежную доставку данных, протокол TCP предусматривает установление

логического соединения, что 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. Благодаря этому протоколу объекты на хосте-отправителе и хосте-получателе могут поддерживать обмен данными в дуплексном режиме. TCP дает возможность без ошибок доставить сформированный на одном из компьютеров поток байтов на любой другой компьютер, входящий в составную сеть.

Второй протокол этого уровня, UDP, является простейшим дейтаграммным протоколом, который используется тогда, когда задача надежного обмена данными либо вообще не ставится, либо решается средствами более высокого уровня — прикладным уровнем или пользовательскими приложениями.

В функции протоколов TCP и UDP входит также исполнение роли связующего звена между прилегающими к транспортному уровню прикладным и сетевым уровнями. От прикладного протокола транспортный уровень принимает задание на передачу данных с тем или иным качеством прикладному уровню-получателю. Нижележащий сетевой уровень протоколы TCP и UDP рассматривают как своего рода инструмент, не очень надежный, но способный перемещать пакет в свободном и рискованном путешествии по составной сети.

Программные модули, реализующие протоколы TCP и UDP, подобно модулям протоколов прикладного уровня, устанавливаются на хостах.

Сетевой уровень, называемый также уровнем Интернета, является стержнем всей архитектуры TCP/IP. Именно этот уровень, функции которого соответствуют сетевому уровню модели OSI, обеспечивает перемещение

пакетов в пределах составной сети, образованной объединением нескольких подсетей. Протоколы сетевого уровня поддерживают интерфейс с вышележащим транспортным уровнем, получая от него запросы на передачу данных по составной сети, а также с нижележащим уровнем сетевых интерфейсов.

Основным протоколом сетевого уровня является межсетевой протокол (Internet Protocol, IP). В его задачу входит продвижение пакета между сетями — от одного маршрутизатора к другому до тех пор, пока пакет не попадет в сеть назначения. В отличие от протоколов прикладного и транспортного уровней, протокол IP развертывается не только на хостах, но и на всех маршрутизаторах (шлюзах). Протокол IP — это дейтаграммный протокол, работающий без установления соединений по принципу доставки с максимальными усилиями. Такой тип сетевого сервиса называют также «ненадежным». К сетевому уровню TCP/IP часто относят протоколы, выполняющие вспомогательные функции по отношению к IP. Это, прежде всего, протоколы маршрутизации RIP и OSPF, предназначенные для изучения топологии сети, определения маршрутов и составления таблиц маршрутизации, на основании которых протокол IP перемещает пакеты в нужном направлении. По этой же причине к сетевому уровню могут быть отнесены протокол межсетевых управляющих сообщений (Internet Control Message Protocol, ICMP), предназначенный для передачи маршрутизатором источнику сведений об ошибках, возникших при передаче пакета, и некоторые другие протоколы.

Идеологическим отличием архитектуры стека TCP/IP от многоуровневой архитектуры других стеков является интерпретация функций самого нижнего уровня — уровня сетевых интерфейсов.

Нижние уровни модели OSI (канальный и физический) реализуют множество функций доступа к среде передачи, формированию кадров, согласованию величин электрических сигналов, кодированию и синхронизации, а также некоторые другие. Все эти весьма конкретные функции составляют суть таких протоколов обмена данными, как Ethernet, PPP и многих других.

У нижнего уровня стека TCP/IP задача существенно проще — он отвечает только за организацию взаимодействия с подсетями разных технологий, входящими в составную сеть. TCP/IP рассматривает любую подсеть, входящую в составную сеть, как средство транспортировки пакетов между двумя соседними маршрутизаторами.

Задачу организации интерфейса между технологией TCP/IP и любой другой технологией промежуточной сети упрощенно можно свести к двум задачам:

- упаковка (инкапсуляция) IP-пакета в единицу передаваемых данных промежуточной сети;
- преобразование сетевых адресов в адреса технологии данной промежуточной сети.

Такой гибкий подход упрощает решение проблемы расширения набора поддерживаемых технологий. При появлении новой популярной технологии она быстро включается в стек TCP/IP путем разработки соответствующего стандарта, определяющего метод инкапсуляции IP-пакетов в ее кадры (например, спецификация RFC 1577, определяющая работу протокола IP через сети ATM, появилась в 1994 году вскоре после принятия основных стандартов ATM). Так как для каждой вновь появляющейся технологии разрабатываются собственные интерфейсные средства, функции этого уровня нельзя определить раз и навсегда, и именно

поэтому нижний уровень стека TCP/IP не регламентируется.

Потоком данных, информационным потоком, или просто потоком, называют данные, поступающие от приложений на вход протоколов транспортного уровня — TCP и UDP.

Протокол TCP «нарезаем» из потока данных сегменты.

Единицу данных протокола UDP часто называют дейтаграммой, или датаграммой. Дейтаграмма — это общее название для единиц данных, которыми оперируют протоколы без установления соединений. К таким протоколам относится и протокол IP, поэтому его единицу данных иногда тоже называют дейтаграммой, хотя достаточно часто используется и другой термин — пакет.

В стеке TCP/IP единицы данных любых технологий, в которые упаковываются IP-пакеты для их последующей передачи через сети составной сети, принято называть также кадрами, или фреймам. При этом не имеет значения, какое название используется для этой единицы данных в технологии составляющей сети. Для TCP/IP фреймом является и кадр Ethernet, и ячейка ATM, и пакет X.25 в тех случаях, когда они выступают в качестве контейнера, в котором IP-пакет переносится через составную сеть.

Типы адресов стека TCP/IP

Итак, для идентификации сетевых интерфейсов используются три типа адресов:

- локальные (аппаратные) адреса;
- сетевые адреса (IP-адреса);
- символьные (доменные) имена

Локальные адреса

В большинстве технологий LAN (Ethernet, FDDI, Token Ring) для однозначной адресации интерфейсов используются MAC-адреса. Существует немало

технологий (X.25, ATM, frame relay), в которых применяются другие схемы адресации. Роль, которую играют эти адреса в TCP/IP, не зависит от того, какая именно технология используется в подсети, поэтому они имеют общее название — локальные (аппаратные) адреса.

Слово «локальный» в контексте TCP/IP означает «действующий не во всей составной сети, а лишь в пределах подсети». Именно в таком смысле понимаются здесь термины: «локальная технология» (технология, на основе которой построена подсеть) и «локальный адрес» (адрес, который используется некоторой локальной технологией для адресации узлов в пределах подсети). Напомним, что в качестве подсети («локальной сети») может выступать сеть, построенная как на основе локальной технологии, например Ethernet, FDDI, так и на основе глобальной технологии, например X.25, Frame Relay. Следовательно, говоря о подсети, мы используем слово «локальная» не как характеристику технологии, на которой построена эта подсеть, а как указание на роль, которую играет эта подсеть в архитектуре составной сети.

Сложности могут возникнуть и при интерпретации определения «аппаратный». В данном случае термин «аппаратный» подчеркивает концептуальное представление разработчиков стека TCP/IP о подсети как о некотором вспомогательном аппаратном средстве, единственной функцией которого является перемещение IP-пакета через подсеть до ближайшего шлюза (маршрутизатора). И не важно, что реально нижележащая локальная технология может быть достаточно сложной, все ее сложности технологией TCP/IP игнорируются.

Рассмотрим, например, случай, когда в составную сеть TCP/IP входит сеть IPX/SPX. Последняя сама может быть разделена на подсети, и так же как IP-сеть, она идентифицирует свои узлы аппаратными и сетевыми IPX-

адресами. Но технология TCP/IP игнорирует многоуровневое строение сети IPX/SPX и рассматривает в качестве локальных адресов узлов подсети IPX/SPX адреса сетевого уровня данной технологии (IPX-адреса). Аналогично, если в составную сеть включена сеть X.25, то локальными адресами узлов этой сети для протокола IP будут соответственно адреса X.25.

Чтобы технология TCP/IP могла решать свою задачу объединения сетей, ей необходима собственная глобальная система адресации, не зависящая от способов адресации узлов в отдельных сетях. Эта система адресации должна позволять универсальным и однозначным способом идентифицировать любой интерфейс составной сети. Очевидным решением является уникальная нумерация всех сетей составной сети, а затем нумерация всех узлов в пределах каждой из этих сетей. Пара, состоящая из номера сети и номера узла, отвечает поставленным условиям и может являться сетевым адресом.

В качестве номера узла может выступать либо локальный адрес этого узла (такая схема принята в стеке IPX/SPX), либо некоторое число, никак не связанное с локальной технологией и однозначно идентифицирующее узел в пределах данной подсети. В первом случае сетевой адрес становится зависимым от локальных технологий, что ограничивает его применение. Например, сетевые адреса IPX/SPX рассчитаны на работу в составных сетях, объединяющих сети, в которых используются только MAC-адреса или адреса аналогичного формата. Второй подход более универсален, он характерен для стека TCP/IP. В технологии TCP/IP сетевой адрес называют IP-адресом.

Если рассматривать IP-сеть, то можно отметить, что маршрутизатор по определению входит сразу в несколько

сетей, следовательно, каждый его интерфейс имеет собственный IP-адрес. Конечный узел также может входить в несколько IP-сетей. В этом случае компьютер должен иметь несколько IP-адресов — по числу сетевых связей. Таким образом, IP-адрес идентифицирует не отдельный компьютер или маршрутизатор, а одно сетевое соединение.

Каждый раз, когда пакет направляется адресату через составную сеть, в его заголовке указывается IP-адрес узла назначения. Но номеру сети назначения каждый очередной маршрутизатор находит IP-адрес следующего маршрутизатора. Перед тем как отправить пакет в следующую сеть, маршрутизатор должен определить на основании найденного IP- адреса следующего маршрутизатора его локальный адрес. Для этой цели протокол IP, обращается к протоколу разрешения адресов (ARP).

Рис. 7.1 — Преобразование адресов

Доменные имена

Для идентификации компьютеров аппаратное и программное обеспечение в сетях TCP / IP полагается на IP-адреса. Например, команда ftp://192.45.66.17 будет устанавливать сеанс связи с нужным IP-сервером, а

команда http://203.23.106.33 откроет начальную страницу на корпоративном веб-сервере. Однако пользователи обычно предпочитают работать с более удобными символьными именами компьютеров.

Символьные идентификаторы сетевых интерфейсов в пределах составной сети строятся по иерархическому принципу. Составляющие полного символьного (или доменного) имени в IP-сетях разделяются точкой и перечисляются в следующем порядке: сначала простое имя хоста, затем имя группы хостов (например, имя организации), потом имя более крупной группы (домена) 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 система доменных имен (Domain Name System, DNS), которая устанавливает это соответствие на основании создаваемых администраторами сети таблиц соответствия. Поэтому доменные имена называют также DNS-именами.

В общем случае сетевой интерфейс может иметь несколько локальных адресов, сетевых адресов и доменных имен.

Формат IP-адреса

В заголовке IP-пакета для хранения IP-адресов отправителя и получателя отводятся два поля, каждое имеет фиксированную длину 4 байта (32 бита). IP-адрес состоит из двух логических частей — номера сети и номера узла в сети.

Наиболее распространенной 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:

10000000 00001010 00000010 00011110

А также в шестнадцатеричном формате:

80.0A.02.1D

Заметим, что запись адреса не предусматривает специального разграничительного знака между номером сети и номером узла. Вместе с тем при передаче пакета по сети часто возникает необходимость разделить адрес на эти две части. Например, маршрутизация, как правило, осуществляется на основании номера сети, поэтому каждый маршрутизатор получая пакет, должен прочитать из соответствующего поля заголовка адрес назначения и выделить из него номер сети. Каким образом маршрутизаторы определяют, какая часть из 32 бит, отведенных под IP-адрес, относится к номеру сети, а какая — к номеру узла.

Можно предложить несколько вариантов решения этой проблемы:

- Простейший из них состоит в использовании фиксированной границы. При этом все 32-битное поле адреса заранее делится на две части не обязательно равной, но фиксированной длины, в одной из которых всегда будет размещаться номер сети, в другой — номер узла. Поскольку поле, которое отводится для хранения номера узла, имеет фиксированную длину, 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 удовлетворять потребности отдельных предприятий и организаций. Именно поэтому он не нашел применения, хотя и использовался на начальном этапе существования технологии TCP/IP (RFC 760).

- Второй подход (RFC 950, RFC 1518) основан на использовании маски, которая позволяет максимально гибко устанавливать границу между номером сети и номером узла. При таком подходе адресное пространство можно использовать для создания множества сетей разного размера.

Маска — это число, применяемое в паре с IP-адресом, причем двоичная запись маски содержит непрерывную последовательность единиц в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети. Граница между последовательностями единиц и нулей в маске соответствует границе между номером сети и номером узла в IP-адресе.

- И, наконец, способ, основанный на классах адресов (RFC 791). Этот способ представляет собой компромисс по отношению к двум предыдущим: размеры сетей хотя и не могут быть произвольными, как при использовании масок, но и не должны быть одинаковыми, как при установлении фиксированных границ. Вводится пять классов адресов: A, B, C, D, E. Три из них — A, B и C — предназначены для адресации сетей, а два — D и E — имеют специальное назначение. Для каждого класса сетевых адресов определено

собственное положение границы между номером сети и номером узла.

Классы IP-адресов

Признаком, на основании которого IP-адрес относят к тому или иному классу, являются значения нескольких первых битов адреса. Таблица ниже иллюстрирует структуру

Таблица 7.1 — Классы IP-адресов.

Класс	Первые биты	Наименьший номер сети	Наибольший номер сети	Максимальное число узлов в сети
A	0	1.0.0.0 (0 — не используется)	126.0.0.0 (127 — зарезервирован)	2^{24}, иоле 3 байта
B	10	128.0.0.0	191.255.0.0	2^{16}, поле 2 байта
C	110	192.0.0.0	223.255.255.0	2^{e}, иоле 1 байт
D	1110	224.0.0.0	239.255.255.255	Групповые адреса
E	11110	240.0.0.0	247.255.255.255	Зарезервировано

- К классу A относится адрес, в котором старший бит имеет значение 0. В адресах класса A под идентификатор сети отводится 1 байт, а остальные 3 байта интерпретируются как номер узла в сети. Сети, все IP-адреса которых имеют значение первого байта в диапазоне от 1 (00000001) до 126(01111110), называются сетями класса A. Значение 0 (00000000) первого байта не используется, а значение 127 (01111111) зарезервировано для специальных целей (см. далее). Сетей класса A сравнительно немного, зато количество узлов в них может достигать 2^{24}, то есть 16 777 216 узлов.
- К классу B относятся все адреса, старшие два бита которых имеют значение 10. В адресах класса B под номер сети и под номер узла отводится по 2 байта. Сети, значения первых двух байтов адресов которых находятся в диапазоне от 128.0 (10000000

00000000) до 191.255 (10111111 11111111), называются сетями класса B. Ясно, что сетей класса B больше, чем сетей класса A, а размеры их меньше. Максимальное количество узлов в сетях класса B составляет 2^{16} (65 536).

- К классу C относятся все адреса, старшие три бита которых имеют значение 110. В адресах класса C под номер сети отводится 3 байта, а под номер узла — 1 байт. Сети, старшие три байта которых находятся в диапазоне от 192.0.0 (11000000 00000000 00000000) до 223.255.255 (11011111 11111111 11111111), называются сетями класса C. Сети класса C наиболее распространены, и наименьшее максимальное число узлов в них равно 2^8 (256).
- Если адрес начинается с последовательности 1110, то он является адресом класса D и обозначает особый групповой адрес (multicast address). В то время как адреса классов A, B и C служат для идентификации отдельных сетевых интерфейсов, то есть являются индивидуальными адресами (unicast address), групповой адрес идентифицирует группу сетевых интерфейсов, которые в общем случае могут принадлежать разным сетям. Интерфейс, входящий в группу, получает наряду с обычным индивидуальным IP-адресом еще один групповой адрес. Если при отправке пакета в качестве адреса назначения указан адрес класса D. то такой пакет должен быть доставлен всем узлам, которые входят в группу.
- Если адрес начинается с последовательности 11110, то это значит, что данный адрес относится к классу E. Адреса этого класса зарезервированы для будущих применений.

Чтобы получить из IP-адреса номер сети и номер узла, требуется не только разделить адрес на две соответствующие части, но и дополнить каждую из них нулями до полных 4 байт. Возьмем, например, адрес класса B 129.64.134.5. Первые два байта идентифицируют сеть, а последующие два — узел. Таким образом, номером сети является адрес 129.64.0.0, а номером узла — адрес 0.0.134.5.

Особые IP-адреса

В TCP/IP существуют ограничения при назначении IP-адресов, а именно номера сетей и номера узлов не могут состоять из одних двоичных нулей или единиц. Отсюда следует, что максимальное количество узлов, приведенное в табл. 7.1 для сетей каждого класса, должно быть уменьшено на 2. Например, в адресах класса C под номер узла отводится 8 бит, которые позволяют задать 256 номеров: от 0 до 255. Однако в действительности максимальное число узлов в сети класса C не может превышать 254, так как адреса 0 и 255 запрещены для адресации сетевых интерфейсов. Из этих же соображений следует, что конечный узел не может иметь адрес типа 98.255.255.255, поскольку номер узла в этом адресе класса A состоит из одних двоичных единиц.

Итак, некоторые IP-адреса интерпретируются особым образом:

- Если IP-адрес состоит только из двоичных нулей, то он называется неопределенным адресом и обозначает адрес того узла, который сгенерировал этот пакет. Адрес такого вида в особых случаях помещается в заголовок IP-пакета в поле адреса отправителя.
- 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. Такой адрес также может быть использован только в качестве адреса отправителя.

- Если все двоичные разряды IP-адреса равны 1, то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета. Такой адрес называется ограниченным широковещательным (limited broadcast). Ограниченность в данном случае означает, что пакет не выйдет за границы данной сети не при каких условиях.
- Если в поле адреса назначения в разрядах, соответствующих номеру узла, стоят только единицы, то пакет, имеющий такой адрес, рассылается всем узлам сети, номер которой указан в адресе назначения. Например, пакет с адресом 192.190.21.255 будет направлен всем узлам сети 192.190.21.0. Такой тип адреса называется широковещательным (broadcast).

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

Особый смысл имеет IP-адрес, первый октет которого равен 127. Этот адрес является внутренним

адресом стека протоколов компьютера (или маршрутизатора). Он используется для тестирования программ, а также для организации работы клиентской и серверной частей приложения, установленных на одном компьютере. Обе программные части данного приложения спроектированы в расчете на то, что они будут обмениваться сообщениями по сети. Но какой же IP-адрес они должны использовать для этого? Адрес сетевого интерфейса компьютера, на котором они установлены? Но это приводит к избыточным передачам пакетов в сеть. Экономичным решением является применение внутреннего адреса 127.0.0.0. В IP-сети запрещается присваивать сетевым интерфейсам IP-адреса, начинающиеся со значения 127. Когда программа посылает данные но IP-адресу 127.x.x.x, то данные не передаются в сеть, а возвращаются модулям верхнего уровня того же компьютера как только что принятые. Маршрут перемещения данных образует «петлю», поэтому этот адрес называется адресом обратной петли (loopback).

Групповые адреса, относящиеся к классу D, предназначены для экономичного распространения в Интернете или большой корпоративной сети аудио- или видеопрограмм, адресованных сразу большой аудитории слушателей или зрителей. Если групповой адрес помещен в поле адреса назначения IP-пакета, то данный пакет должен быть доставлен сразу нескольким узлам, которые образуют группу с номером, указанным в поле адреса. Один и тот же узел может входить в несколько групп. В общем случае члены группы могут распределяться по различным сетям, находящимся друг от друга на произвольно большом расстоянии. Групповой адрес не делится на номера сети и узла и обрабатывается маршрутизатором особым образом. Основное назначение

групповых адресов — распространение информации но схеме «один ко многим».

Использование масок при IP-адресации

Снабжая каждый IP-адрес маской, можно отказаться от понятий классов адресов и сделать более гибкой систему адресации.

Пусть, например, для IP-адреса 129.64.134.5 указана маска 255.255.128.0, то есть в двоичном виде IP-адрес 129.64.134.5 — это:

10000001.01000000.10000110.00000101, а маска 255.255.128.0 в двоичном виде выглядит так: 11111111.11111111.10000000.00000000. Если игнорировать маску и интерпретировать адрес 129.64.134.5 на основе классов, то номером сети является 129.64.0.0, а номером узла — 0.0.134.5 (поскольку адрес относится к классу B).

Если же использовать маску, то 17 последовательных двоичных единиц в маске 255.255.128.0, «наложенные» на IP-адрес 129.64.134.5, делят его на две части, номер сети: 10000001.01000000.1 и номер узла:

0000110.00000101.

В десятичной форме записи номера сети и узла, дополненные нулями до 32 бит. выглядят соответственно как 129.64.128.0 и 0.0.6.5.

Наложение маски можно интерпретировать как выполнение логической операции И (AND). Так, в предыдущем примере номер сети из адреса 129.64.134.5 является результатом выполнения логической операции AND с маской 255.255.128.0: 10000001 01000000 10000110 00000101 AND

11111111.11111111.10000000.00000000 Для стандартных классов сетей маски имеют следующие значения:

- класс A — 11111111.00000000.00000000.00000000 (255.0.0.0);
- класс B — 11111111.11111111.00000000.00000000 (255.255.0.0);
- класс C — 11111111.11111111 11111111.00000000(255.255.255.0).

Для записи масок используются и другие форматы. Например, удобно интерпретировать значение маски, записанной в шестнадцатеричном коде: FF.FF.00.00 — маска для адресов класса B. Еще чаще встречается обозначение 185.23.44.206/16 — данная запись говорит о том, что маска для этого адреса содержит 16 единиц или что в указанном IP-адресе под номер сети отведено 16 двоичных разрядов.

Механизм масок широко распространен в маршрутизации IP, причем маски могут использоваться для самых разных целей. С их помощью администратор может разбивать одну, выделенную ему поставщиком услуг сеть определенного класса на несколько других, не требуя от него дополнительных номеров сетей — эта операция называется разделением на подсети (subnetting). На основе этого же механизма поставщики услуг могут объединять адресные пространства нескольких сетей путем введения так называемых «префиксов» с целью уменьшения объема таблиц маршрутизации и повышения за счет этого производительности маршрутизаторов — такая операция называется объединением подсетей (supernetting). Подробнее об этом мы поговорим при изучении технологии бесклассовой междоменной маршрутизации.

Порядок назначения IP-адресов

По определению схема IP-адресации должна обеспечивать уникальность нумерации сетей, а также уникальность нумерации узлов в пределах каждой из сетей. Следовательно, процедуры назначения номеров как сетям, так и узлам сетей должны быть централизованными. Рекомендуемый порядок назначения IP-адресов дается в спецификации RFC 2050.

Назначение адресов автономной сети

Когда дело касается сети, являющейся частью Интернета, уникальность нумерации может быть обеспечена только усилиями специально созданных для этого центральных органов. В небольшой же автономной IP-сети условие уникальности номеров сетей и узлов может быть выполнено силами сетевого администратора.

В этом случае в распоряжении администратора имеется все адресное пространство, так как совпадение IP-адресов в не связанных между собой сетях не вызовет никаких отрицательных последствий. Администратор может выбирать адреса произвольным образом, соблюдая лишь синтаксические правила и учитывая ограничения на особые адреса. (Таким образом, номер узла в технологии TCP/IP назначается независимо от его локального адреса.) Однако при таком подходе исключена возможность в будущем подсоединить данную сеть к Интернету. Действительно, произвольно выбранные адреса данной сети могут совпасть с централизовано назначенными адресами Интернета. Для того чтобы избежать коллизий, связанных с такого рода совпадениями, в стандартах Интернета определено несколько диапазонов так называемых частных адресов, рекомендуемых для автономного использования:

- в классе A — сеть 10.0.0.0;
- в классе B — диапазон из 16 номеров сетей (172.16.0.0-172.31.0.0);

- в классе С - диапазон из 255 сетей (192.168.0.0-192.168.255.0).

Эти адреса, исключенные из множества централизованно распределяемых, составляют огромное адресное пространство, достаточное для нумерации узлов автономных сетей практически любых размеров. Заметим также, что частные адреса, как и при произвольном выборе адресов, в разных автономных сетях могут совпадать. В то же время использование частных адресов для адресации автономных сетей делает возможным корректное подключение их к Интернету. Применяемые при этом специальные технологии подключения исключают коллизии адресов.

Централизованное распределение адресов

В больших сетях, подобных Интернету, уникальность сетевых адресов гарантируется централизованной, иерархически организованной системой их распределения. Номер сети может быть назначен только по рекомендации специального подразделения Интернета. Главным органом регистрации глобальных адресов в Интернете с 1998 года является неправительственная некоммерческая организация ICANN (Internet Corporation for Assigned Names and Numbers). 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, а те, в свою очередь, распределяют их между своими клиентами, среди которых могут быть и более мелкие поставщики. Проблемой централизованного распределения адресов является их дефицит. Уже сравнительно давно очень трудно получить адрес класса В и практически невозможно стать обладателем адреса класса А. При этом

надо отметить, что дефицит обусловлен не только ростом сетей, но и тем, что имеющееся адресное пространство используется нерационально. Очень часто владельцы сетей класса С расходуют лишь небольшую часть из имеющихся у них 254 адресов. Рассмотрим пример, когда две сети необходимо соединить глобальной связью. В таких случаях в качестве линии связи используют два маршрутизатора, соединенных по двухточечной схеме. Для вырожденной сети, образованной линией связи, связывающей порты двух смежных маршрутизаторов, приходится выделять отдельный номер сети, хотя в этой сети всего два узла.

Для смягчения проблемы дефицита адресов разработчики стека TCP/IP предлагают разные подходы. Принципиальным решением является переход на новую версию протокола IP — протокол IPv6, в котором резко расширяется адресное пространство. Однако и текущая версия протокола IP (IPv4) поддерживает технологии, направленные на более экономное расходование IP-адресов, такие, например, как NAT и CIDR.

Адресация и технология CIDR

Технология бесклассовой междоменной маршрутизации (Classless Inter-Domain Routing, CIDR), которая описана в документах RFC 1517, RFC 1518, RFC 1519, RFC 1520 и о которой впервые было официально объявлено в 1993 году, позволяет центрам распределения адресов избежать выдачи абонентам излишних адресов.

Деление IP-адреса на номера сети и узла в технологии CIDR происходит на основе маски переменной длины, назначаемой поставщиком услуг. Непременным условием применимости CIDR является наличие у организации, распоряжающейся адресами, непрерывных диапазонов адресов. Такие адреса имеют одинаковый

префикс, то есть одинаковую цифровую последовательность в нескольких старших разрядах.

Благодаря CIDR поставщик услуг получает возможность «нарезать» блоки из выделенного ему адресного пространства в соответствии с действительными требованиями каждого клиента.

Отображение IP-адресов на локальные адреса

Одной из главных задач, которая ставилась при создании протокола IP, являлось обеспечение совместной согласованной работы в сети, состоящей из подсетей, в общем случае использующих разные сетевые технологии. Взаимодействие технологии TCP/IP с локальными технологиями подсетей происходит многократно при перемещении IP-пакета по составной сети. На каждом маршрутизаторе протокол IP определяет, какому следующему маршрутизатору в этой сети надо направить пакет. В результате решения этой задачи протоколу IP становится известен IP-адрес интерфейса следующего маршрутизатора (или конечного узла, если эта сеть является сетью назначения). Чтобы локальная технология сети смогла доставить пакет на следующий маршрутизатор, необходимо:

- упаковать пакет в кадр соответствующего для данной сети формата (например, Ethernet);
- снабдить данный кадр локальным адресом следующего маршрутизатора. Решением этих задач, как уже отмечалось, занимается уровень сетевых интерфейсов стека TCP/IP.

Система DNS

Плоские символьные имена

В операционных системах, которые первоначально разрабатывались для локальных сетей, таких как Novell NetWare, Microsoft Windows или IBM OS/2, пользователи всегда работали с символьными именами компьютеров.

Так как локальные сети состояли из небольшого числа компьютеров, применялись так называемые плоские имена, состоящие из последовательности символов, не разделенных на части. Примерами таких имен являются: NW1_1, rnail2, MOSCOW_SALES_2. Для установления соответствия между символьными именами и MAC-адресами в этих операционных системах применялся механизм широковещательных запросов, подобный механизму запросов протокола ARP. Так, широковещательный 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.

Иерархические символьные имена

В стеке TCP/IP применяется доменная система имен, которая имеет иерархическую древовидную структуру, допускающую наличие в имени произвольного количества составных частей.

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 отделяются друг от друга точкой.

Например, в имени home.microsoft.com составляющая home является именем одного из компьютеров в домене microsoft.com.

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, один человек может нести ответственность за то, чтобы все имена с окончанием «ru» имели уникальную следующую вниз по иерархии часть. То есть все имена типа www.ru, mail.mmt.ru или m2.zil.mmt.ru отличаются 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 имен (domain). Например, имена www.zil.mmt.ru,ftp.zil.mmt.ru, yandex.ru и sl.mgu.ru входят в домен ru, так как все они имеют одну общую старшую часть — имя ru. Другим примером является домен mgu.ru. Из представленных имен в него входят имена s1.mgu.ru, s2.mgu.ru и rn.mgu.ru. Этот домен образуют имена, у которых две старшие части равны mgu.ru. Администратор домена mgu.ru несет ответственность за уникальность имен следующего уровня, входящих в домен, то есть имен s1, s2 и rm. Образованные домены s1.mgu.ru, s2.mgu.ru и rn.mgu.ru являются поддоменами домена mgu.ru, так как имеют общую старшую часть имени. Часто проддомены для

краткости называют только младшей частью имени, то есть в нашем случае поддоменами являются s1, s2 и rn.

Термин «домен» очень многозначен, поэтому его нужно трактовать в рамках определенного контекста. Помимо доменов имен стека TCP/IP в компьютерной литературе часто упоминаются домены Windows NT, домены коллизий и некоторые другие. Общим у всех этих терминов является то, что они описывают некоторое множество компьютеров, обладающее каким-либо определенным свойством.

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

По аналогии с файловой системой в доменной системе имен различают краткие, относительные и полные доменные имена. Краткое доменное имя — это имя конечного узла сети: хоста или порта маршрутизатора. Краткое имя — это лист дерева имен. Относительное доменное 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.

Компьютеры, имена которых относятся к одному и тому же домену, могут иметь абсолютно независимые друг от друга IP-адреса, принадлежащие различным сетям и подсетям. Например, в домен mgu.ru могут входить хосты с адресами 132.13.34.15, 201.22.100.33 и 14.0.0.6.

Корневой домен управляется центральными органами Интернета, в частности уже упоминавшейся нами организацией ICANN. Домены верхнего уровня назначаются для каждой страны, а также для различных

типов организаций. Имена этих доменов должны следовать международному стандарту ISO 3166. Для обозначения стран используются трехбуквенные и двухбуквенные аббревиатуры, например ru (Россия), uk (Великобритания), fi (Финляндия), us (Соединенные Штаты), а для различных типов организаций, например, 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 — сетевые организации (например, nsf.net).

Каждый домен администрирует отдельная организация, которая обычно разбивает свой домен на поддомены и передает функции администрирования этих поддоменов другим организациям. Чтобы получить доменное имя, необходимо зарегистрироваться в какой-либо организации, которой делегированы полномочия по распределению имен доменов.

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

Схема работы DNS

Широковещательный способ установления соответствия между символьными именами и локальными адресами, подобный протоколу ARP, хорошо работает только в небольшой локальной сети, не разделенной на подсети. В крупных сетях, где возможность всеобщей широковещательной рассылки не поддерживается, нужен другой способ разрешения символьных имен. Хорошей

альтернативой широковещательной рассылке является применение централизованной службы, поддерживающей соответствие между различными типами адресов всех компьютеров сети. Например, компания Microsoft для своей корпоративной операционной системы Windows NT разработала централизованную службу WINS, которая поддерживала базу данных NetBIOS-имен и соответствующих им IP-адресов.

В сетях TCP/IP соответствие между доменными именами и IP-адресами может устанавливаться средствами как локального хоста, так и централизованной службы.

На раннем этапе развития Интернета на каждом хосте вручную создавался текстовый файл с известным именем hosts.txt. Этот файл состоял из некоторого количества строк, каждая из которых содержала одну пару «доменное имя — IP-адрес», например:

rtiino.acme.com — 102.54.94.97 По мере роста Интернета файлы hosts.txt также увеличивались в объеме, и создание масштабируемого решения для разрешения имен стало необходимостью. Таким решением стала централизованная служба DNS (Domain Name System — система доменных имен), основанная на распределенной базе отображений «доменное имя — IP- адрес». Служба DNS использует в своей работе DNS-серверы и DNS-клиенты. DNS-серверы поддерживают распределенную базу отображений, а DNS-клиенты обращаются к серверам с запросами об отображении разрешении доменного имени на IP-адрес. Служба DNS использует текстовые файлы почти такого же формата, как и файл hosts, и эти файлы администратор также подготавливает вручную. Однако служба DNS опирается на иерархию доменов, и каждый DNS-сервер хранит только часть имен сети, а не все имена, как это происходит при использовании файлов hosts. При росте количества узлов в сети проблема масштабирования

решается созданием новых доменов и пол доменов имен и добавлением в службу DNS новых серверов.

Для каждого домена имен создается свой DNS-сервер. На серверах применяют два подхода к распределению имен. В первом случае сервер может хранить отображения «доменное имя — IP-адрес» для всего домена, включая все его поддомены. Однако такое решение оказывается плохо масштабируемым, так как при добавлении новых поддоменов нагрузка на этот сервер может превысить его возможности. Чаще используется другой 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. Например, в первом случае DNS-сервер домена mmt.ru будет хранить отображения для всех имен, оканчивающихся на mmt.ru (wwwl.zil.mmt.ru, ftp.zil.mmt.ru, mail.mmt.ru и т.д.). Во втором случае этот сервер хранит отображения только имен типа mail.mmt.ru, www.mmt.ru, а все остальные отображения должны храниться на DNS-сервере поддомена zil.

Каждый DNS-сервер помимо таблицы отображений имен содержит ссылки на DNS-серверы 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 широко известны (их можно узнать, например, в InterNIC).

Процедура разрешения DNS-имени во многом аналогична процедуре поиска файловой системой адреса файла по его символьному имени. Действительно, в обоих случаях составное имя отражает иерархическую структуру организации соответствующих справочников — каталогов файлов или DNS-таблиц. 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 проверяются 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 отличием файловой системы от службы DNS является то, что первая расположена на одном компьютере, а вторая по своей природе является распределенной.

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

1. DNS-клиент обращается к корневому DNS-серверу с указанием полного доменного имени.

2. DNS-сервер отвечает клиенту, указывая адрес следующего DNS-сервера, обслуживающего домен верхнего уровня, заданный в следующей старшей части запрошенного имени.

3. DNS-клиент делает запрос следующего DNS-сервера, который отсылает его к DNS-серверу нужного поддомена и т.д., пока не будет найден DNS-сервер, в

котором хранится соответствие запрошенного имени IP-адресу. Этот сервер дает окончательный ответ клиенту.

Такая процедура разрешения имени называется нерекурсивной, когда клиент сам итеративно выполняет последовательность запросов к разным серверам имен. Эта схема загружает клиента достаточно сложной работой, и она применяется редко.

Во втором варианте реализуется рекурсивная процедура:

1. DNS-клиент запрашивает локальный DNS-сервер, то есть тот сервер, обслуживающий поддомен, которому принадлежит имя клиента.

2. Далее возможны два варианта действий:

- если локальный DNS-сервер знает ответ, то он сразу же возвращает его клиенту (это может произойти, когда запрошенное имя входит в тот же поддомен, что и имя клиента, или 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 делал клиент в предыдущем варианте, а получив ответ, передает его клиенту, который все это время просто ждет его от своего локального DNS-сервера.

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

Для ускорения поиска IP-адресов DNS-серверы широко применяют кэширование проходящих через них ответов. Чтобы служба DNS могла оперативно отрабатывать изменения, происходящие в сети, ответы

кэшируются на относительно короткое время — обычно от нескольких часов до нескольких дней.

Обратная зона

Служба DNS предназначена не только для нахождения IP-адреса по имени хоста, но и для решения обратной задачи — нахождению DNS-имени по известному IP-адресу.

Многие программы и утилиты, пользующиеся службой DNS, пытаются найти имя узла по его адресу в том случае, когда пользователем задан только адрес (или этот адрес программа узнала из пришедшего пакета). Обратная запись не всегда существует даже для тех адресов, для которых есть прямые записи. Ее могут просто забыть создать или же ее создание требует дополнительной оплаты. Обратная задача решается в Интернете путем организации так называемых обратных зон.

Обратная зона — это система таблиц, которая хранит соответствие между IP-адресами и DNS-имена хостов некоторой сети. Для организации распределенной службы и использования для поиска имен того же программного обеспечения, что и для поиска адресов, применяется оригинальный подход, связанный с представлением IP-адреса в виде DNS- имени.

Первый этап преобразования заключается в том, что составляющие IP-адреса интерпретируются как составляющие DNS-имени. Например, адрес 192.31.106.0 рассматривается как состоящий из старшей части, соответствующей домену 192, затем идет домен 31, в который входит домен 106.

Далее, учитывая, что при записи IP-адреса старшая часть является самой левой частью адреса, а при записи DNS-имени — самой правой, то составляющие в преобразованном адресе указываются в обратном порядке,

то есть для данного примера — 106.31.192. Для хранения соответствия всех адресов, начинающихся, например, с числа 192, заводится зона 192 со своими серверами имен. Для записей о серверах, поддерживающих старшие в иерархии обратные зоны, создана специальная зона in-addr.arpa, поэтому полная запись для использованного в примере адреса выглядит так: 106.31.192.in-addr.arpa.

Серверы для обратных зон используют файлы баз данных, не зависящие от файлов основных зон, в которых имеются записи о прямом соответствии тех же имен и адресов. Такая организация данных может приводить к несогласованности, так как одно и то же соответствие вводится в файлы дважды.

Протокол DHCP

Для нормальной работы сети каждому сетевому интерфейсу компьютера и маршрутизатора должен быть назначен IP-адрес.

Процедура присвоения адресов происходит в ходе конфигурирования компьютеров и маршрутизаторов. Назначение IP-адресов может происходить вручную в результате выполнения процедуры конфигурирования интерфейса, для компьютера сводящейся, например, к заполнению системы экранных форм. При этом администратор должен помнить, какие адреса из имеющегося множества он уже использовал для других интерфейсов, а какие еще свободны. При конфигурировании помимо IP-адресов сетевых интерфейсов (и соответствующих масок) устройству сообщается ряд других конфигурационных параметров. При конфигурировании администратор должен назначит!» клиенту не только IP-адрес, но и другие параметры стека TCP/IP, необходимые для его эффективной работы, например маску и IP-адрес маршрутизатора по умолчанию, IP-адрес DNS-сервера, доменное имя компьютера и т. п.

Даже при не очень большом размере сети эта работа представляет для администратора утомительную процедуру.

Протокол динамического конфигурирования хостов (Dynamic Host Configuration Protocol, DHCP) автоматизирует процесс конфигурирования сетевых интерфейсов, обеспечивая отсутствие дублирования адресов за счет централизованного управления их распределением. Работа DHCP описана в RFC 2131 и 2132.

Режимы DHCP

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, содержащее IP-адрес и некоторые другие конфигурационные параметры.

При этом сервер DHCP может работать в разных режимах, включая:

- ручное назначение статических адресов;
- автоматическое назначение статических адресов;
- автоматическое распределение 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, то есть имеют одно и то же значение в поле номера сети.

В ручном режиме администратор, помимо пула доступных адресов, снабжает DHCP-сервер информацией о жестком соответствии IP-адресов физическим адресам или другим идентификаторам клиентских узлов. DHCP-сервер, пользуясь этой информацией, всегда выдаст определенному DHCP-клиенту один и тот же назначенный

ему администратором IP-адрес (а также набор других конфигурационных параметров).

В режиме автоматического назначения статических адресов DHCP-сервер самостоятельно без вмешательства администратора произвольным образом выбирает клиенту IP-адрес из пула наличных IP-адресов. Адрес дается клиенту из пула в постоянное пользование, то есть между идентифицирующей информацией клиента и его IP-адресом по-прежнему, 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 сроком аренды. Когда компьютер, являющийся DHCP-клиентом, удаляется из подсети, назначенный ему IP-адрес автоматически освобождается. Когда компьютер подключается к другой подсети, то ему автоматически назначается новый адрес. Ни пользователь, ни сетевой администратор не вмешиваются в этот процесс. Это 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, режим динамического распределения адресов в принципе позволяет строить IP-сеть, количество узлов в которой превышает количество имеющихся в распоряжении администратора IP-адресов.

Рассмотрим преимущества, которые дает динамическое распределение пула адресов на примере организации, в которой сотрудники значительную часть рабочего времени проводят вне офиса — дома или в

командировках. Каждый из них имеет портативный компьютер, который во время пребывания в офисе подключается к корпоративной IP-сети. Возникает вопрос, сколько IP-адресов необходимо этой организации?

Первый ответ — столько, скольким сотрудникам необходим доступ в сеть. Если их 500 человек, то каждому из них должен быть назначен IP-адрес и выделено рабочее место. То есть администрация должна получить у поставщика услуг адреса двух сетей класса C и оборудовать соответствующим образом помещение. Однако вспомним, что сотрудники в этой организации редко появляются в офисе, значит, большая часть ресурсов при таком решении будет простаивать.

Второй ответ — столько, сколько сотрудников обычно присутствует в офисе (с некоторым запасом). Если обычно в офисе работает не более 50 сотрудников, то достаточно получить у поставщика услуг пул из 64 адресов и установить в рабочем помещении сеть с 64-я коннекторами для подключения компьютеров. Но возникает другая проблема — кто и как будет конфигурировать компьютеры, состав которых постоянно меняется?

Существует два пути. Во-первых, администратор (или сам мобильный пользователь) может конфигурировать компьютер вручную каждый раз, когда возникает необходимость подключения к офисной сети. Такой подход требует от администратора (или пользователей) большого объема рутинной работы, следовательно — это плохое решение. Гораздо привлекательнее выглядят возможности автоматического динамического назначения DHCP-адресов. Действительно, администратору достаточно один раз при настройке DHCP-сервера указать диапазон из 64 адресов, а каждый вновь прибывающий мобильный пользователь будет

просто физически подключать в сеть свой компьютер, на котором запускается DHCP-клиент.

Он запросит конфигурационные параметры и автоматически получит их от DHCP-сервера. Таким образом, для работы 500 мобильных сотрудников достаточно иметь в офисной сети 64 IP-адреса и 64 рабочих места.

Алгоритм динамического назначения адресов

Администратор управляет процессом конфигурирования сети, определяя два основных конфигурационных параметра DHCP-сервера: пул адресов, доступных распределению, и срок аренды. Срок аренды диктует, как долго компьютер может использовать назначенный IP-адрес, перед тем как снова запросить его от DHCP-сервера. Срок аренды зависит от режима работы пользователей сети. Если это небольшая сеть учебного заведения, куда со своими компьютерами приходят многочисленные студенты для выполнения лабораторных работ, то срок аренды может быть равен длительности лабораторной работы. Если же это корпоративная сеть, в которой сотрудники предприятия работают на регулярной основе, то срок аренды может быть достаточно длительным — несколько дней или даже недель. DHCP-сервер должен находиться в одной подсети с клиентами, учитывая, что клиенты посылают ему широковещательные запросы. Для снижения риска выхода сети из строя из-за отказа DHCP-сервера в сети иногда ставят резервный DHCP-сервер.

Иногда наблюдается и обратная картина: в сети нет ни одного DHCP-сервера. В этом случае его подменяет связной DHCP-агент — программное обеспечение, играющее роль посредника между DHCP-клиентами и DHCP-серверами (пример такого варианта — сеть 2). Связной агент переправляет запросы клиентов из одной

сети DHCP-серверу из другой сети. Таким образом, один DHCP-сервер может обслуживать DHCP-клиентов нескольких разных сетей.

Вот как выглядит упрощенная схема обмена сообщениями между клиентскими и серверными частями DHCP.

1. Когда компьютер включают, установленный на нем DHCP-клиент посылает ограниченное широковещательное сообщение DHCP-поиска (IP-пакет с адресом назначения, состоящим из одних единиц, который должен быть доставлен всем узлам данной IP- сети).

2. Находящиеся в сети DHCP-серверы получают это сообщение. Если в сети DHCP- серверы отсутствуют, то сообщение DHCP-поиска получает связной DHCP-агент. Он пересылает это сообщение в другую, возможно, значительно отстоящую от него сеть DHCP-серверу, IP-адрес которого ему заранее известен.

3. Все DHCP-серверы, получившие сообщение DHCP-поиска, посылают DHCP-клиенту, обратившемуся с запросом, свои DHCP-предложения. Каждое предложение содержит IP-адрес и другую конфигурационную информацию. (DHCP-сервер, находящийся в другой сети, посылает ответ через агента.)

4. DHCP-клиент собирает конфигурационные DHCP-предложения от всех DHCP- серверов. Как правило, он выбирает первое из поступивших предложений и отправляет в сеть широковещательный DHCP-запрос. В этом запросе содержатся идентификационная информация о DHCP-сервере, предложение которого принято, а также значения принятых конфигурационных параметров.

5. Все DHCP-серверы получают DHCP-запрос, и только один выбранный DHCP-сервер посылает положительную DHCP-квитанцию (подтверждение IP-адреса и параметров аренды), а остальные серверы

аннулируют свои предложения, в частности возвращают в свои пулы предложенные адреса.

6. DHCP-клиент получает положительную DHCP-квитанцию и переходит в рабочее состояние.

Время от времени компьютер пытается обновить параметры аренды у DHCP-сервера. Первую попытку он делает задолго до истечения срока аренды, обращаясь к тому серверу, от которого он получил текущие параметры. Если ответа нет или ответ отрицательный, он через некоторое время снова посылает запрос. Так повторяется несколько раз, и если все попытки получить параметры у того же сервера оказываются безуспешными, клиент обращается к другому серверу. Если и другой сервер отвечает отказом, то клиент теряет свои конфигурационные параметры и переходит в режим автономной работы. Также DHCP-клиент может по своей инициативе досрочно отказаться от выделенных ему параметров.

В сети, где адреса назначаются динамически, нельзя быть уверенным в адресе, который в данный момент имеет тот или иной узел. И такое непостоянство IP-адресов влечет за собой некоторые проблемы.

Во-первых, возникают сложности при преобразовании символьного доменного имени в IP- адрес. Действительно, представьте себе функционирование системы DNS, которая должна поддерживать таблицы соответствия символьных имен IP-адресам в условиях, когда последние меняются каждые два часа! Учитывая это обстоятельство, для серверов, к которым пользователи часто обращаются по символьному имени, назначают статические IP-адреса, оставляя динамические только для клиентских компьютеров. Однако в некоторых сетях количество серверов настолько велико, что их ручное конфигурирование становится слишком обременительным.

Эго привело к разработке усовершенствованной версии DNS (так называемой динамической системы DNS), в основе которой лежит согласование информационной адресной базы в службах DHCP и DNS.

Во-вторых, трудно осуществлять удаленное управление и автоматический мониторинг интерфейса (например, сбор статистики), если в качестве его идентификатора выступает динамически изменяемый IP-адрес.

Наконец, для обеспечения безопасности сети многие сетевые устройства могут блокировать (фильтровать) пакеты, определенные ноля которых имеют некоторые заранее заданные значения. Другими словами, при динамическом назначении адресов усложняется фильтрация пакетов по IP-адресам.

Последние две проблемы проще всего решаются отказом от динамического назначения адресов для интерфейсов, фигурирующих в системах мониторинга и безопасности.

Список используемых источников

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для ВУЗов. 4-е изд. - СПб.:Питер, 2010. - 944 с.:ил.
2. Информационные технологии. Материал из Википедии — свободной энциклопедии [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/Информационные_технологии (дата обращения 11.11.2016).
3. Технология Ethernet. Материалы из Википедии — Свободной энциклопедии
4. Статья Технологии локальных сетей. [Электронный ресурс]. — Режим доступа: http://sernam.ru/book_icn.php?id=15 (дата обращения 11.11.2016).
5. Лекция Ethernet [Электронный ресурс] — Режим доступа: https://refdb.ru/look/2351219.html (дата обращения 11.11.2016).
6. Сети Token Ring Технология [Электронный ресурс] — Режим доступа: http://life-prog.ru/view_zam2.php?id=70 (дата обращения 11.11.2016).
7. Статья Ethernet vs. Token Ring [Электронный ресурс] — Режим доступа: http://www.osp.ru/cw/1997/32/22983/ (дата обращения 11.11.2016).
8. Основы технологии FDDI [Электронный ресурс] — Режим доступа: http://citforum.ru/nets/lvs/glava_12.shtml (дата обращения 11.11.2016).
9. Статья «Что такое система доменных имен (DNS) и как она работает» [Электронный ресурс] — Режим

доступа: http://site.nic.ru/content/view/225/29 (дата обращения 11.11.2016).

10. Статья DHCP. Материал из Википедии — свободной энциклопедии [Электронный ресурс] — Режим доступа: https://ru.wikipedia.org/wiki/DHCP (дата обращения 11.11.2016).