Основы компьютерных сетей

Основные понятия

Компьютерные сети, называемые также сетями передачи данных, являются логическим результатом эволюции двух важнейших научно-технических отраслей современной цивилизации - вычислительной техники и телекоммуникационных технологий.

Интернет - это компьютерная сеть, которая связывает между собой сотни миллионов вычислительных устройств по всему миру

В корпоративных и университетских городках и все в большей степени в домашних условиях для подключения конечных систем к граничным маршрутизаторам используются локальные вычислительные сети или ЛВС (Local Area Networks, или LAN)

Протокол передачи данных - набор соглашений интерфейса логического уровня, которые определяют обмен данными между различными программами.

Бит в секунду, бит/с (англ. bits per second, bps) - базовая единица измерения скорости передачи информации, используемая на физическом уровне сетевой модели OSI или TCP/IP.

Кадр (англ. frame) - фрагмент данных протокола канального уровня модели OSI, передаваемый по линии связи.

Протокольный блок данных (PDU) - это термин взаимодействия открытых систем (OSI), используемый в телекоммуникациях, который относится к группе информации, добавляемой или удаляемой уровнем модели OSI.

Пакет - (англ. packet) - это сегмент данных отправляемых с одного компьютера или устройства на другое, по сети.

TCP (Transmission Control Protocol - протокол управления передачей), является надежным протоколом с установлением соединений, позволяющим без ошибок доставлять байтовый поток с одной машины на любую другую машину объединенной сети

UDP (User Datagram Protocol - протокол пользовательских дейтограмм), является ненадежным протоколом без установления соединения, не использующим последовательное управление потоком протокола TCP, а предоставляющим свое собственное

Датаграмма (англ. datagram, дейтаграмма) - блок информации, передаваемый протоколом через сеть связи без предварительного установления соединения и создания виртуального канала.

Протоколы:

ARP (англ. Address Resolution Protocol - протокол определения адреса) - протокол в компьютерных сетях, предназначенный для определения MAC-адреса другого компьютера по известному IP-адресу.

Internet Protocol (IP, досл. «межсетевой протокол») - маршрутизируемый протокол сетевого уровня стека TCP/IP. Именно IP стал тем протоколом, который объединил отдельные компьютерные сети во всемирную сеть Интернет. Неотъемлемой частью протокола является адресация сети

GRE (англ. Generic Routing Encapsulation - общая инкапсуляция маршрутов) - протокол туннелирования сетевых пакетов, разработанный компанией Cisco Systems. Его основное назначение - инкапсуляция пакетов сетевого уровня сетевой модели OSI в IP-пакеты.

ICMP (англ. Internet Control Message Protocol - протокол межсетевых управляющих сообщений) - сетевой протокол, входящий в стек протоколов TCP/IP. В основном ICMP используется для передачи сообщений об ошибках и других исключительных ситуациях,

возникших при передаче данных, например, запрашиваемая услуга недоступна или хост, или маршрутизатор не отвечают.

gRPC (Remote Procedure Calls) - это система удалённого вызова процедур (RPC) с открытым исходным кодом, первоначально разработанная в Google в 2015 году.

WebSocket - протокол связи поверх TCP-соединения, предназначенный для обмена сообщениями между браузером и веб-сервером, используя постоянное соединение.

HTTP (англ. HyperText Transfer Protocol - «протокол передачи гипертекста») - протокол прикладного уровня передачи данных, изначально - в виде гипертекстовых документов в формате HTML, в настоящее время используется для передачи произвольных данных.

HTTPS (аббр. от англ. HyperText Transfer Protocol Secure) - расширение протокола HTTP для поддержки шифрования в целях повышения безопасности.

SSL (англ. Secure Sockets Layer - уровень защищённых сокетов) - криптографический протокол, который подразумевает более безопасную связь. Он использует асимметричную криптографию для аутентификации ключей обмена, симметричное шифрование для сохранения конфиденциальности, коды аутентификации сообщений для целостности сообщений.

TLS (англ. transport layer security - Протокол защиты транспортного уровня), как и его предшественник SSL (англ. secure sockets layer - слой защищённых сокетов), - криптографические протоколы, обеспечивающие защищённую передачу данных между узлами в сети Интернет. TLS и SSL используют асимметричное шифрование для аутентификации, симметричное шифрование для конфиденциальности и коды аутентичности сообщений для сохранения целостности сообщений.

DHCP (англ. Dynamic Host Configuration Protocol - протокол динамической настройки узла) - сетевой протокол, позволяющий сетевым устройствам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP.

SSH (англ. Secure Shell - «безопасная оболочка») - сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений (например, для передачи файлов).

TELNET (сокр. от англ. Teletype Network) - сетевой протокол для реализации текстового терминального интерфейса по сети (в современной форме - при помощи транспорта TCP). Название «telnet» имеют также некоторые утилиты, реализующие клиентскую часть протокола.

Назначение сетей:

Локальные сети

Локальными сетями называют частные сети, размещающиеся, как правило, в одном здании или на территории какой-либо организации. Их часто используют для объединения компьютеров и рабочих станций в офисах компании или предприятия бытовой электроники для предоставления совместного доступа к ресурсам (например, принтерам) и обмена информацией. Когда локальные сети используются предприятиями, их называют сеть предприятия (enterprise networks). Беспроводные ЛВС сейчас очень популярны, особенно в домах, более старых офисных зданиях, кафетериях и других местах, где слишком сложно провести кабели. В этих системах у каждого компьютера есть радиомодем и антенна, которую он использует, чтобы общаться с другими компьютерами. В большинстве случаев каждый компьютер говорит с устройством в потолке. Это устройство, названное AP (Точка доступа), беспроводный маршрутизатор, или базовая станция, передает пакеты между беспроводными компьютерами, а также между ними и Интернетом. Точка доступа похожа на популярного ребенка в школе, потому что все хотят говорить с ним. Однако, если другие

компьютеры достаточно близки, они могут общаться непосредственно друг с другом в конфигурации соединения равноправных узлов ЛВС
В проводных ЛВС используются различные технологии передачи. Большинство из них использует медные провода, а некоторые - оптоволокно. ЛВС ограничены в размере, это означает, что максимальное время передачи ограничено и известно заранее. Знание этих границ помогает с задачей разработки сетевых протоколов. Как правило, проводные ЛВС работают на скоростях от 100 Мбит/с до 1 Гбит/с, имеют низкую задержку (микросекунды или наносекунды) и делают очень немного ошибок. Более новые ЛВС могут работать со скоростью 10 Гбит/с. По сравнению с беспроводными сетями проводные ЛВС превышают их по всем параметрам работы. Послать сигналы по проводу или через волокно проще, чем по воздуху

Глобальные сети
Глобальная сеть (wide area network, WAN) охватывает значительную географическую область, часто целую страну или даже континент. Мы начнем разговор о них с проводных глобальных сетей, используя в качестве примера компанию, имеющую подразделения в разных городах. Сеть, соединяет офисы, находящиеся в Перте, Мельбурне и Брисбене. Каждый из них содержит компьютеры, предназначенные для выполнения программ пользователя (то есть приложений). Мы будем следовать традиционной терминологии и называть эти машины хостами. Хосты соединяются коммуникационными подсетями, называемыми для краткости просто подсетями. Задачей подсети является передача сообщений от хоста хосту, подобно тому, как телефонная система переносит слова (то есть просто звуки) от говорящего слушающему
В большинстве глобальных сетей подсеть состоит из двух раздельных компонентов: линий связи и переключающих элементов. Линии связи переносят данные от машины к машине. Они могут представлять собой медные провода, оптоволокно или даже радиосвязь. Большинство компаний не имеют собственных линий связи, поэтому они арендуют их у телекоммуникационной компании. Переключающие элементы являются специализированными компьютерами, используемыми для соединения двух или более линий связи. Когда данные появляются на входной линии, переключающий элемент должен выбрать выходную линию для дальнейшего маршрута этих данных. В прошлом для названия этих компьютеров не было стандартной терминологии. Сейчас их называют маршрутизаторами (router), однако читателю следует знать, что по поводу терминологии в данном случае единого мнения не существует.
Большинство глобальных сетей содержит большое количество кабелей или телефонных линий, соединяющих пару маршрутизаторов. Если какие-либо два маршрутизатора не связаны линией связи напрямую, то они должны общаться при помощи других маршрутизаторов. В сети может быть много путей, которые соединяют эти два маршрутизатора. Метод принятия решения называется алгоритмом маршрутизации. Существует много таких алгоритмов. То, как каждый маршрутизатор принимает решение, куда послать пакет, называется алгоритмом пересылки.

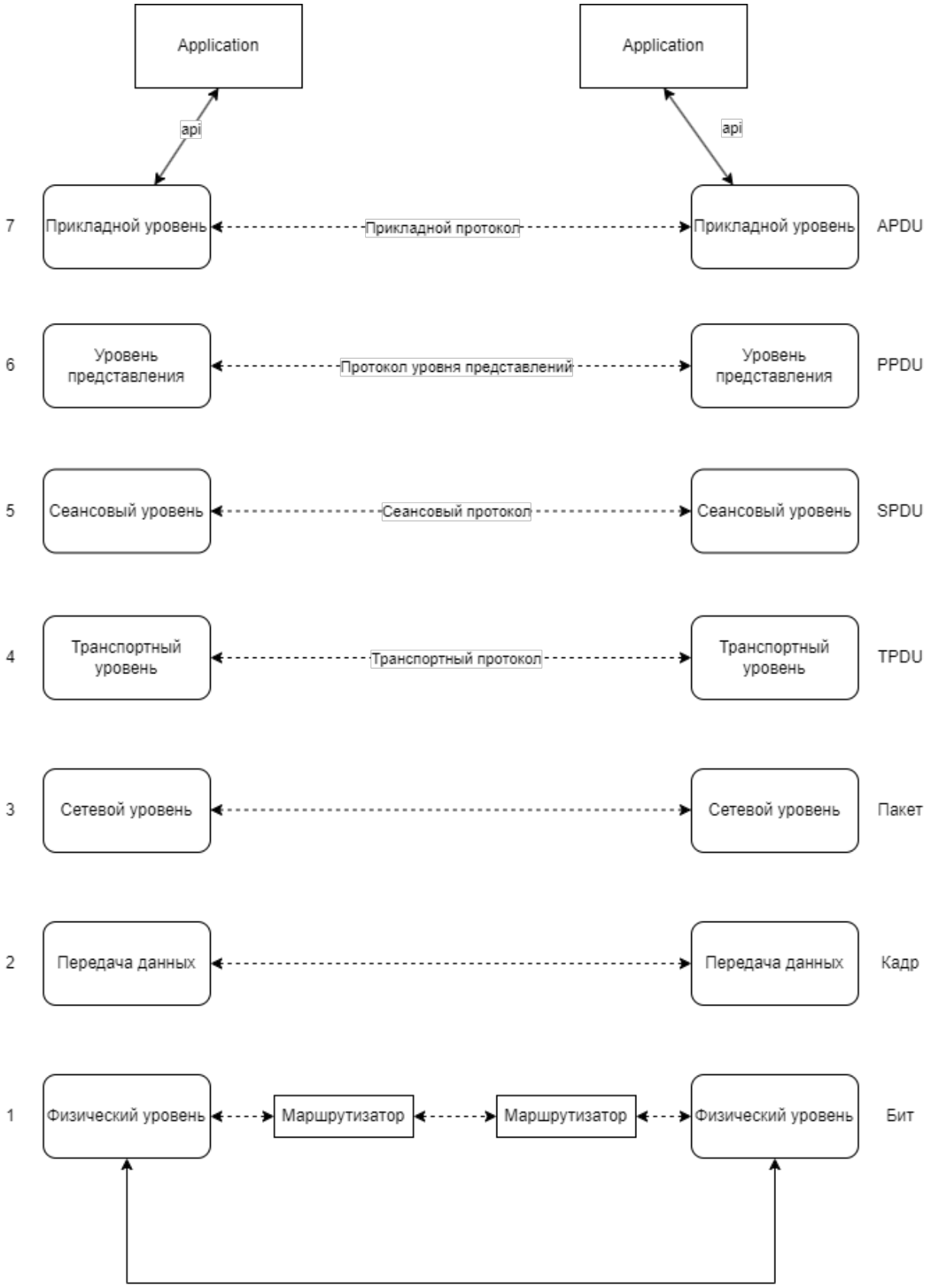
Эталонные модели

Обсудив многоуровневые сети в общих чертах, пора рассмотреть несколько примеров. Мы опишем два важных архитектурных типа - эталонные модели OSI и TCP/IP. Несмотря на то что протоколы, связанные с эталонной моделью OSI, сейчас не используются, сама модель до сих пор весьма актуальна, а свойства ее уровней, которые будут обсуждаться в этом разделе, очень важны. В эталонной модели TCP/IP все наоборот: сама модель сейчас почти не используется, а ее протоколы являются едва ли не самыми

распространенными. Исходя из этого, мы обсудим подробности, касающиеся обеих моделей. К тому же иногда приходится больше узнавать из поражений, чем из побед

Эталонная модель OSI

Эталонная модель OSI (за исключением физической среды). Эта модель основана на разработке Международной организации по стандартизации (International Organization for Standardization, ISO) и является первым шагом к международной стандартизации протоколов, используемых на различных уровнях (Day и Zimmerman затем она была пересмотрена в 1995 году (Day, 1995). Называется эта структура эталонной моделью взаимодействия открытых систем ISO (ISO OSI (Open System Interconnection) Reference Model), поскольку она связывает открытые системы, то есть системы, открытые для связи с другими системами. Для краткости мы будем называть эту модель просто «модель OSI». Модель OSI имеет семь уровней. Появление именно такой структуры было обусловлено следующими соображениями. 1. Уровень должен создаваться по мере необходимости отдельного уровня абстракции. 2. Каждый уровень должен выполнять строго определенную функцию. 3. Выбор функций для каждого уровня должен осуществляться с учетом создания стандартизированных международных протоколов. 4. Границы между уровнями должны выбираться так, чтобы поток данных между интерфейсами был минимальным. 5. Количество уровней должно быть достаточно большим, чтобы различные функции не объединялись в одном уровне без необходимости, но не слишком высоким, чтобы архитектура не становилась громоздкой. Ниже мы обсудим каждый уровень модели, начиная с самого нижнего. Обратите внимание: модель OSI не является сетевой архитектурой, поскольку она не описывает службы и протоколы, используемые на каждом уровне. Она просто определяет, что должен делать каждый уровень. Тем не менее ISO также разработала стандарты для каждого уровня, хотя эти стандарты не входят в саму эталонную модель. Каждый из них был опубликован как отдельный международный стандарт. Эта модель (частично) широко используется, хотя связанные с ней протоколы долго были забыты.



Физический уровень

Физический уровень занимается реальной передачей необработанных битов по каналу связи. При разработке сети необходимо убедиться, что когда одна сторона передает единицу, то принимающая сторона получает также единицу, а не ноль. Принципиальными вопросами здесь являются следующие: какое напряжение должно использоваться для отображения единицы, а какое для нуля; сколько микросекунд длится бит; может ли передача

производиться одновременно в двух направлениях; как устанавливается начальная связь и как она прекращается, когда обе стороны закончили свои задачи; из какого количества проводов должен состоять кабель и какова функция каждого провода. Вопросы разработки в основном связаны с механическими, электрическими и процедурными интерфейсами, а также с физическим носителем, лежащим ниже физического уровня

Уровень передачи данных

Основная задача уровня передачи данных - быть способным передавать «сырые» данные физического уровня по надежной линии связи, свободной от необнаруженных ошибок, и маскировать реальные ошибки, так что сетевой уровень их не видит. Эта задача выполняется при помощи разбиения входных данных на кадры, обычный размер которых колеблется от нескольких сот до нескольких тысяч байт. Кадры данных передаются последовательно с обработкой кадров подтверждения, отсылаемых обратно получателем. Еще одна проблема, возникающая на уровне передачи данных (а также и на большей части более высоких уровней), - как не допустить ситуации, когда быстрый передатчик заваливает приемник данными. Может быть предусмотрен некий механизм регуляции, который информировал бы передатчик о наличии свободного места в буфере приемника на текущий момент. В широковещательных сетях существует еще одна проблема уровня передачи данных: как управлять доступом к совместно используемому каналу. Эта проблема разрешается введением специального дополнительного подуровня уровня передачи данных - подуровня доступа к носителю

Сетевой уровень

Сетевой уровень занимается управлением операциями подсети. Важнейшим моментом здесь является определение маршрутов пересылки пакетов от источника к пункту назначения. Маршруты могут быть жестко заданы в виде таблиц и редко меняться либо, что бывает чаще, автоматически изменяться, чтобы избегать отказавших компонентов. Кроме того, они могут задаваться в начале каждого соединения, например, терминальной сессии, такого как подключения к удаленной машине. Наконец, они могут быть в высокой степени динамическими, то есть вычисляемыми заново для каждого пакета с учетом текущей загруженности сети. Если в подсети одновременно присутствует слишком большое количество пакетов, то они могут закрыть дорогу друг другу, образуя заторы в узких местах. Недопущение подобной закупорки также является задачей сетевого уровня в соединении с более высокими уровнями, которые адаптируют загрузку. В более общем смысле, сетевой уровень занимается предоставлением определенного уровня сервиса (это касается задержек, времени передачи, вопросов синхронизации). При путешествии пакета из одной сети в другую также может возникнуть ряд проблем. Так, способ адресации, применяемый в одной сети, может отличаться от принятого в другой. Сеть может вообще отказаться принимать пакеты из-за того, что они слишком большого размера. Также могут различаться протоколы и т. д. Именно сетевой уровень должен разрешать все эти проблемы, позволяя объединять разнородные сети. В широковещательных сетях проблема маршрутизации очень проста, поэтому в них сетевой уровень очень примитивный или вообще отсутствует.

Транспортный уровень

Основная функция транспортного уровня - принять данные от сеансового уровня, разбить их при необходимости на небольшие части, передать их сетевому уровню и гарантировать, что эти части в правильном виде прибудут по назначению. Кроме того, все это должно быть сделано эффективно и таким образом, чтобы изолировать более высокие уровни от каких-либо изменений в аппаратной технологии с течением времени. Транспортный уровень также определяет тип сервиса, предоставляемого сеансовому уровню и, в конечном счете, пользователям сети. Наиболее популярной разновидностью транспортного соединения

является защищенный от ошибок канал между двумя узлами, поставляющий сообщения или байты в том порядке, в каком они были отправлены. Однако транспортный уровень может предоставлять и другие типы сервисов, например пересылку отдельных сообщений без гарантии соблюдения порядка их доставки или одновременную отправку сообщения различным адресатам по принципу широковещания. Тип сервиса определяется при установке соединения. (Строго говоря, полностью защищенный от ошибок канал создать совершенно невозможно. Говорят лишь о таком канале, уровень ошибок в котором достаточно мал, чтобы им можно было пренебречь на практике.) Транспортный уровень является настоящим сквозным уровнем, то есть доставляющим сообщения от источника адресату. Другими словами, программа на машине-источнике поддерживает связь с подобной программой на другой машине при помощи заголовков сообщений и управляющих сообщений. На более низких уровнях для поддержки этого соединения устанавливаются соединения между всеми соседними машинами, через которые проходит маршрут сообщений. Различие между уровнями с 1-го по 3-й, действующих по принципу звеньев цепи, и уровнями с 4-го по 7-й, являющимися сквозными.

Сеансовый уровень

Сеансовый уровень позволяет пользователям различных компьютеров устанавливать сеансы связи друг с другом. При этом предоставляются различные типы сервисов, среди которых управление диалогом (отслеживание очередности передачи данных), управление маркерами (предотвращение одновременного выполнения критичной операции несколькими системами) и синхронизация (установка служебных меток внутри длинных сообщений, позволяющих продолжить передачу с того места, на котором она оборвалась, даже после сбоя и восстановления).

Уровень представления

В отличие от более низких уровней, задача которых - достоверная передача битов и байтов, уровень представления занимается по большей части синтаксисом и семантикой передаваемой информации. Чтобы было возможно общение компьютеров с различными внутренними представлениями данных, необходимо преобразовывать форматы данных друг в друга, передавая их по сети в неком стандартизированном виде. Уровень представления занимается этими преобразованиями, предоставляя возможность определения и изменения структур данных более высокого уровня (например, записей баз данных).

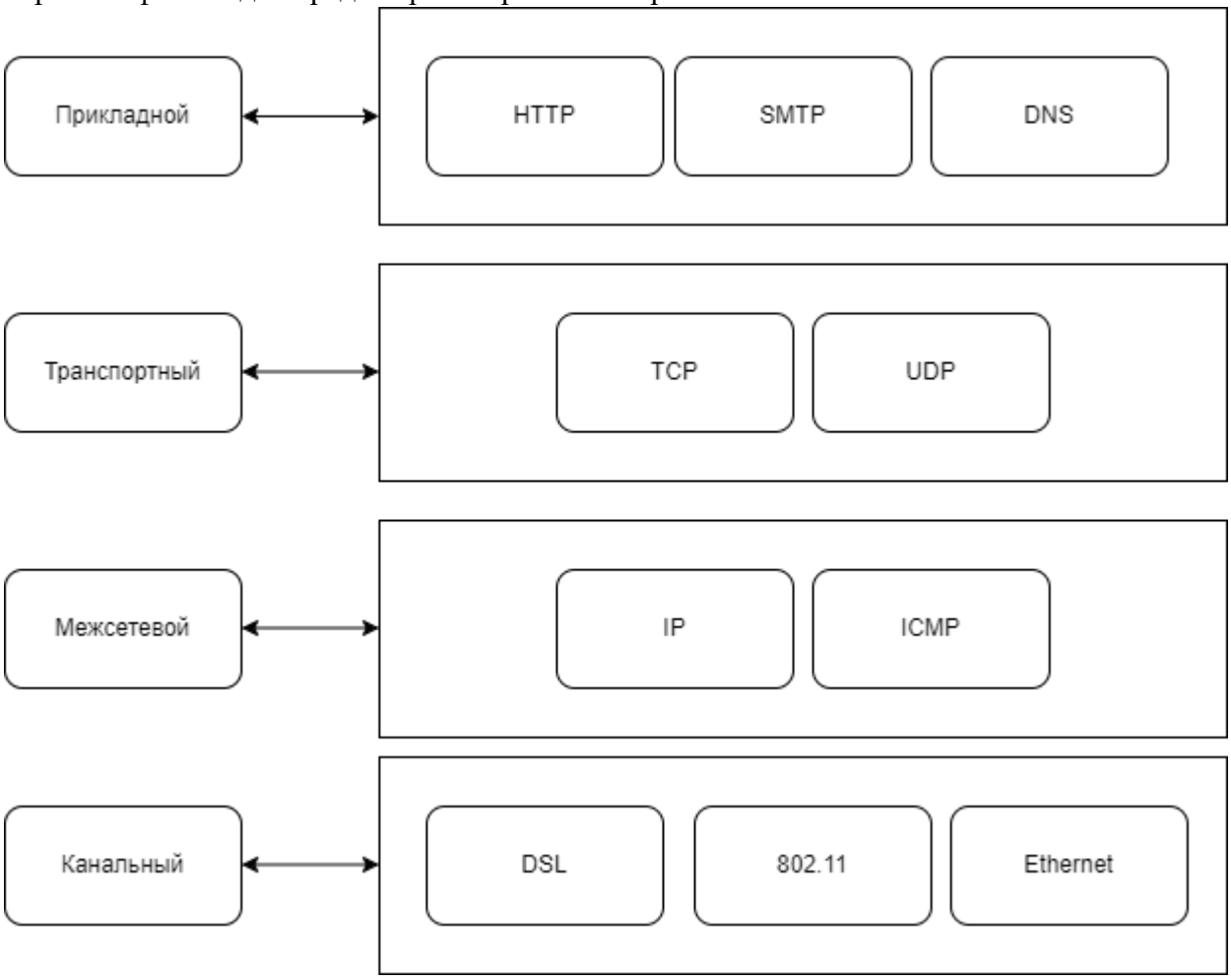
Прикладной уровень

Прикладной уровень содержит набор популярных протоколов, необходимых пользователям. Одним из наиболее распространенных является протокол передачи гипертекста HTTP (HyperText Transfer Protocol), который составляет основу технологии Всемирной паутины. Когда браузер запрашивает веб-страницу, он передает ее имя (адрес) и рассчитывает на то, что сервер, на котором расположена страница, будет использовать HTTP. Сервер в ответ отсылает страницу. Другие прикладные протоколы используются для передачи файлов, электронной почты, сетевых рассылок.

Эталонная модель TCP/IP

Рассмотрим теперь эталонную модель, использовавшуюся в компьютерной сети ARPANET, которая является бабушкой нынешних сетей, а также в ее наследнице, всемирной сети Интернет. Хотя краткую историю сети ARPANET мы рассмотрим чуть позднее, некоторые ключевые моменты ее следует отметить прямо сейчас. ARPANET была исследовательской сетью, финансируемой Министерством обороны США. В конце концов, она объединила сотни университетов и правительственных зданий при помощи выделенных телефонных линий. Когда впоследствии появились спутниковые сети и радиосети, возникли большие

проблемы при объединении с ними других сетей с помощью имеющихся протоколов. Понадобилась новая эталонная архитектура. Таким образом, возможность объединять различные сети в единое целое являлась одной из главных целей с самого начала. Позднее эта архитектура получила название эталонной модели TCP/IP, в соответствии со своими двумя основными протоколами. Первое ее описание встречается в книге Cerf и Kahn (1974), позднее превращается в стандарт (Braden, 1989). Конструктивные особенности модели обсуждаются в издании Clark, 1988. Поскольку Министерство обороны США беспокоилось, что ценные хосты, маршрутизаторы и межсетевые шлюзы могут быть мгновенно уничтожены, другая важная задача состояла в том, чтобы добиться способности сети сохранять работоспособность при возможных потерях подсетевого оборудования, так чтобы при этом связь не прерывалась. Другими словами, Министерство обороны США требовало, чтобы соединение не прерывалось, пока функционируют приемная и передающая машины, даже если некоторые промежуточные машины или линии связи внезапно вышли из строя. Кроме того, от архитектуры нужна была определенная гибкость, поскольку предполагалось использовать приложения с различными требованиями, от переноса файлов до передачи речи в реальном времени.



Канальный уровень

Все эти требования привели к выбору сети с пакетной коммутацией, основанной на уровне без установления соединения, который работает в различных сетях. Самый низкий уровень в модели, уровень канала, описывает то, как и что каналы, такие как последовательные линии и классический Ethernet, должны сделать, чтобы удовлетворить потребности этого межсетевого уровня без установления соединения. Это на самом деле не уровень вообще, в нормальном смысле слова, а скорее интерфейс между каналами передачи и узлами. В ранних материалах о модели TCP/IP мало что об этом говорится.

Межсетевой уровень

Все эти требования обусловили выбор модели сети с коммутацией пакетов, в основе которой лежал не имеющий соединений межсетевой уровень. Примерно соответствует сетевому уровню в OSI. Этот уровень, называемый интернет-уровнем или межсетевым уровнем, является основой всей архитектуры. Его задача заключается в обеспечении возможности каждого хоста посылать пакеты в любую сеть и независимо двигаться к пункту назначения (например, в другой сети). Они могут прибывать совершенно в другом порядке, чем были отправлены. Если требуется соблюдение порядка отправления, эту задачу выполняют более верхние уровни. Обратите внимание, что слово «интернет» здесь используется в своем первоначальном смысле, несмотря на то что этот уровень присутствует в сети Интернет. Здесь можно увидеть аналогию с почтовой системой. Человек может бросить несколько международных писем в почтовый ящик в одной стране, и, если повезет, большая часть из них будет доставлена по правильным адресам в других странах. Вероятно, письма по дороге пройдут через несколько международных почтовых шлюзов, однако это останется тайной для корреспондентов. В каждой стране (то есть в каждой сети) могут быть свои марки, свои предпочитаемые размеры конвертов и правила доставки, незаметные для пользователей почтовой службы. Межсетевой уровень определяет официальный формат пакета и протокол IP, с дополнительным протоколом ICMP (Internet Control message Protocol, межсетевой протокол управления сообщениями). Задачей межсетевого протокола является доставка IP-пакетов к пунктам назначения. Основными аспектами здесь являются выбор маршрута пакета и недопущение закупорки транспортных артерий (хотя IP не оказался эффективным для избегания скоплений).

Транспортный уровень

Уровень, расположенный над межсетевым уровнем модели TCP/IP, как правило, называют транспортным. Он создан для того, чтобы объекты одного ранга на приемных и передающих хостах могли поддерживать связь, подобно транспортному уровню модели OSI. На этом уровне должны быть описаны два сквозных протокола. Первый, TCP (Transmission Control Protocol - протокол управления передачей), является надежным протоколом с установлением соединений, позволяющим без ошибок доставлять байтовый поток с одной машины на любую другую машину объединенной сети. Он разбивает входной поток байтов на отдельные сообщения и передает их межсетевому уровню. На пункте назначения получающий TCP-процесс собирает из полученных сообщений выходной поток. Кроме того, TCP осуществляет управление потоком, чтобы быстрый отправитель не завалил информацией медленного получателя. Второй протокол этого уровня, UDP (User Dataram Protocol - протокол пользовательских дейтограмм2), является ненадежным протоколом без установления соединения, не использующим последовательное управление потоком протокола TCP, а предоставляющим свое собственное. Он также широко используется в одноразовых клиент-серверных запросах и приложениях, в которых оперативность важнее аккуратности, например при передаче речи и видео. Взаимоотношения протоколов IP, TCP и UDP. Со времени создания протокола IP этот протокол был реализован во многих других сетях.

Прикладной уровень

В модели TCP/IP нет сеансового уровня и уровня представления. В этих уровнях просто не было необходимости, поэтому они не были включены в модель. Вместо этого приложения просто включают все функции сеансов и представления, которые им нужны. Опыт работы с моделью OSI доказал правоту этой точки зрения: большинство приложений мало нуждаются в этих уровнях. Над транспортным уровнем располагается прикладной уровень. Он содержит все протоколы высокого уровня. К старым протоколам относятся протокол

виртуального терминала (TELNET), протокол переноса файлов (FTP) и протокол электронной почты (SMTP). С годами было добавлено много других протоколов. Некоторые наиболее важные, которые мы рассмотрим. Это DNS (Domain Name Service - служба имен доменов), позволяющая преобразовывать имена хостов в сетевые, HTTP, протокол, используемый для создания страниц на World Wide Web, а также RTP, протокол для представления мультимедиа в реальном времени, таких как звук или фильмы.

Сравнение эталонных моделей OSI и TCP

У моделей OSI и TCP имеется много общих черт. Обе модели основаны на концепции стека независимых протоколов. Функциональность уровней также во многом схожа. Например, в обеих моделях уровни, начиная с транспортного и выше, предоставляют сквозную, не зависящую от сети транспортную службу для процессов, желающих обмениваться информацией. Эти уровни образуют поставщика транспорта. Также в каждой модели уровни выше транспортного являются прикладными потребителями транспортных сервисов. Несмотря на это фундаментальное сходство, у этих моделей имеется и ряд отличий. В данном разделе мы рассмотрим ключевые различия. Обратите внимание на то, что мы сравниваем именно эталонные модели, а не соответствующие им стеки протоколов. Сами протоколы будут обсуждаться несколько позднее. Книга (Piscitello и Chapin, 1993) целиком посвящена сравнению моделей TCP/IP и OSI. Для модели OSI центральными являются три концепции. 1. Службы. 2. Интерфейсы. 3. Протоколы. Вероятно, наибольшим вкладом модели OSI стало явное разделение этих трех концепций. Каждый уровень предоставляет некоторые сервисы для расположенного выше уровня. Сервис определяет, что именно делает уровень, но не то, как он это делает и каким образом объекты, расположенные выше, получают доступ к данному уровню. Интерфейс уровня определяет способ доступа к уровню для расположенных выше процессов. Он описывает параметры и ожидаемый результат. Он также ничего не сообщает о внутреннем устройстве уровня. Наконец, равноранговые протоколы, применяемые в уровне, являются внутренним делом самого уровня. Для выполнения поставленной ему задачи (то есть предоставления сервиса) он может использовать любые протоколы. Кроме того, уровень может менять протоколы, не затрагивая работу приложений более высоких уровней. Эти идеи очень хорошо соответствуют современным идеям объектно-ориентированного программирования. Уровень может быть представлен в виде объекта, обладающего набором методов (операций), к которым может обращаться внешний процесс. Семантика этих методов определяет набор служб, предоставляемых объектом. Параметры и результаты методов образуют интерфейс объекта. Внутреннее устройство объекта можно сравнить с протоколом уровня. За пределами объекта оно никого не интересует и никому не видно. Изначально в модели TCP/IP не было четкого разделения между службами, интерфейсом и протоколами, хотя и производились попытки изменить это, чтобы сделать ее более похожей на модель OSI. Так, например, единственными настоящими сервисами, предоставляемыми межсетевым уровнем, являются SEND IP PACKET (послать IPпакет) и RECEIVE IP PACKET (получить IP-пакет). В результате в модели OSI протоколы скрыты лучше, чем в модели TCP/IP, и при изменении технологии они могут быть относительно легко заменены. Возможность проводить подобные изменения, не затрагивая другие уровни, является одной из главных целей многоуровневых протоколов. Эталонная модель OSI была разработана прежде, чем были изобретены протоколы для нее. Такая последовательность событий означала, что эта модель не была настроена на какой-то конкретный набор протоколов, что делало ее универсальной. Обратной стороной такого порядка действий было то, что у разработчиков было мало опыта в данной области и не было четкого представления о том,

какие функции должен выполнять каждый уровень. Например, уровень передачи данных изначально работал только в сетях с передачей от узла к узлу. С появлением широковещательных сетей в модель потребовалось ввести новый подуровень. В дальнейшем, когда на базе модели OSI начали строить реальные сети с использованием существующих протоколов, обнаружилось, что они не соответствуют требуемым спецификациям служб. Поэтому в модель пришлось добавить подуровни для устранения несоответствия. Наконец, изначально ожидалось, что в каждой стране будет одна сеть, управляемая правительством и использующая протоколы OSI, поэтому никто и не думал об объединении различных сетей. В действительности все оказалось не так. С моделью TCP/IP было все наоборот: сначала появились протоколы, а уже затем была создана модель, описывающая существующие протоколы. Таким образом, не было проблемы с соответствием протоколов модели. Они ей соответствовали прекрасно. Единственной проблемой было то, что модель не соответствовала никаким другим стекам протоколов. В результате она не использовалась для описания каких-нибудь других сетей, отличных от TCP/IP. Если взглянуть на эти две модели поближе, то, прежде всего, обратит на себя внимание различие в количестве уровней: в модели OSI семь уровней, в модели TCP/ IP - четыре. В обеих моделях имеются межсетевой, транспортный и прикладной уровни, а остальные уровни различные. Еще одно различие между моделями лежит в сфере возможности использования связи на основе соединений и связи без установления соединения. Модель OSI на сетевом уровне поддерживает оба типа связи, а на транспортном уровне - только связь на основе соединений (поскольку транспортные службы являются видимыми для пользователя). В модели TCP/IP на сетевом уровне есть только один режим связи (без установления соединения), но на транспортном уровне она поддерживает оба режима, предоставляя пользователям выбор. Этот выбор особенно важен для простых протоколов запрос-ответ

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 друг на друга, например сетевого IP-адреса на локальный, доменного имени - на 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 иных решений.

Ключевым словом, которое характеризует принятый в TCP/IP подход к решению этих проблем, является **масштабируемость.** Процедуры, предлагаемые TCP/IP для назначения, отображения и конфигурирования адресов, одинаково хорошо работают в сетях разного масштаба.

Типы адресов стека TCP/IP:

Для идентификации сетевых интерфейсов используются три типа адресов:

- локальные (аппаратные) адреса;
- сетевые адреса (1P-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 все 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» в терминологии TCP/IP) может выступать сеть, построенная как на основе технологии LAN, например Ethernet, FDDI, так и на основе технологии WAN, 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 (в настоящее время представляющий поучительный исторический пример), 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.

Сетевые IP-адреса

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, или в терминологии TCP/ IP - **IP-адресом.**

Глядя на топологическую схему IP-сети, можно отметить, что маршрутизатор по определению входит сразу в несколько сетей, следовательно, каждый его интерфейс должен иметь собственный IP-адрес. Конечный узел, имеющий несколько сетевых интерфейсов, также может входить в несколько IP-сетей, а значит, иметь несколько IP-адресов по числу сетевых связей. Таким образом, подчеркнем еще раз - IP-адрес идентифицирует не отдельный узел сети (компьютер или маршрутизатор), а одно сетевое соединение, или, что одно и то же в данном контексте, один сетевой интерфейс.

Каждый раз, когда пакет направляется адресату через составную сеть, в его заголовке указывается IP-адрес узла назначения. По 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. Между IP-адресом и локальным адресом узла нет никакой функциональной зависимости, поэтому единственный способ установления соответствия - ведение таблицы. Эту задачу решает протокол разрешения адресов ARP

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 ftp-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 pc1.prod.example.ru

Символьные имена называют также доменными именами.

Между IP-адресом узла и его доменным именем (так же, как и локальным адресом) нет никакой функциональной зависимости, поэтому для установления соответствия требуются таблицы. В сетях TCP/IP используется специальная сетевая служба, называемая **системой доменных имен** (Domain Name System, **DNS),** которая автоматически устанавливает соответствие между доменными именами и IP-адресами на основании создаваемых администраторами сети таблиц соответствия. По этой причине доменные имена называют также **DNS-именами.**

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

Формат IP-адреса.

В заголовке IP-пакета предусмотрены поля для хранения IP-адреса отправителя и IP- адреса получателя. Каждое из этих полей имеет фиксированную длину 4 байта (32 бита). Как уже было сказано, 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. Решение очень простое, но хорошее ли? 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.

Второй подход основан на применении *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. Например, если маска, связываемая с некоторым IP-адресом, имеет вид 11111111111100000000000000000000, то номеру сети соответствуют 10 старших разрядов в двоичном представлении данного IP-адреса.

И наконец, способ, основанный на **классах адресов.** 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

Класс	**Первые биты**	**Наименьший номер сети**	**Наибольший номер сети**	**Максимальное число узлов в сети**
A	**0**	**1.0.0.0**	**126.0.0.0**	**2^24, поле 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**	**2^8, поле 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**

Исходя из приведенной структуры адресов и информации из таблицы, можно сделать несколько очевидных выводов. Сетей класса A сравнительно немного, зато количество узлов в них очень большое, оно может достигать 2^24, что равно 16 777 216 узлов. Сетей класса B больше, чем сетей класса A, но их размеры меньше, максимальное количество узлов в сетях класса B составляет 2^16 (65 536). Сетей класса C больше всего, но они характеризуются самым маленьким максимально возможным количеством узлов, всего 2^8(256).

В то время как адреса классов A, B и C используются для идентификации отдельных сетевых интерфейсов, то есть являются **индивидуальными адресами** (unicast address), **групповые адреса** (multicast address) класса D идентифицируют 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. Адрес класса D начинается с последовательности 1110 .

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 четырех байтов. 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, для сетей каждого класса, должно быть уменьшено на 2. Например, в адресах класса С под номер узла отводится 8 бит, которые позволяют задать 256 номеров: от 0 до 255. Однако в действительности максимальное число узлов в сети класса С 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 А состоит из одних двоичных единиц.

Введя эти ограничения, разработчики технологии TCP/IP получили возможность расширить функциональность системы адресации следующим 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 подсети ни 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).

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 по 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).

Уже упоминавшиеся *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 по схеме «один ко многим». От того, найдут ли групповые адреса широкое применение (сейчас их используют в основном небольшие

экспериментальные «островки» в Интернете), зависит, сможет ли Интернет создать серьезную конкуренцию радио и телевидению.

Использование масок при IP-адресации

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

Пусть, например, для 1P-адреса 129.64.134.5 указана маска 255.255.128.0, то есть в двоичном виде 1P-адрес 129.64.134.5 равен:

10000001.01000000.10000110.00000101,

В то время как маска 255.255.128.0 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;
- 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 записи масок используются и другие форматы. Например, удобно интерпретировать значение маски, записанной в шестнадцатеричном коде: FE.FF.00.00 - маска для адресов класса B. Еще чаще встречается запись с префиксом 185.23.44.206/26 - данная запись говорит о том, что маска для этого адреса содержит 26 единиц.

Класс адресов	Десятичная форма	Двоичная форма	Шестнадцатеричная форма	Префикс
A	255.0.0.0	11111111.00000000.00000000.00000000	FF.00.00.00	/8
B	255.255.0.0	11111111.11111111.00000000.00000000	FF.FF.00.00	/16
C	255.255.255.0	11111111.11111111.11111111.00000000	FF.FF.FF.00	/24

Механизм масок широко распространен в маршрутизации IP, причем маски могут использоваться для самых разных целей. С их помощью администратор может разбить 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 «вручную» сетевым администратором.

В этом случае в распоряжении администратора имеется все адресное пространство, так как совпадение IP-адресов в не связанных между собой сетях не вызовет никаких отри цательных последствий. Администратор может выбирать адреса произвольным образом, соблюдая лишь синтаксические правила и учитывая ограничения на особые 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 C —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 их корректное подключение к Интернету. Применяемые при этом специальные технологии подключения исключают коллизии адресов1.

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 B и практически невозможно стать обладателем адреса класса A. 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 C 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) основана на использовании масок для более гибкого распределения адресов и более эффективной маршрутизации. Она допускает произвольное разделение IP-адреса на поля для нумерации сети и узлов. При такой системе адресации клиенту может быть выдан пул адресов, более точно соответствующий его запросу, чем это происходит при адресации, основанной на классах адресов.

Например, если *клиенту А* требуется всего 13 адресов, то вместо выделения ему сети стандартного класса С (класса с наименьшим числом узлов - 256) ему может был назначен пул адресов 193.20.30.0/28. Эта запись, имеющая вид *IP -адрес/маска*, интерпретируется следующим образом: «сеть, не принадлежащая ни к какому стандартному классу, номер которой содержится в 28 старших двоичных разрядах 1Р-адреса 193.20.30.0, имеющая 4-битовое поле для нумерации 16 узлов». Все это вполне удовлетворяет требованиям клиента А. Очевидно, что такой вариант намного более экономичен, чем раздача сетей стандартных классов «целиком».

Определение пула адресов в виде пары IP-адрес/маска возможно только при выполнении нескольких условий. Прежде всего адресное пространство, из которого организация, распределяющая адреса, «нарезает» адресные пулы для заказчиков, должно быть *непрерывным.* При таком условии все адреса имеют общий **префикс** - одинаковую последовательность цифр в старших разрядах адреса.

Пусть, например, провайдер располагает адресами в диапазоне 193.20.0.0-193.23.255.255, или в десятичной записи:

11000001**.**00010100.00000000.00000000 - 11000001**.**00010111.11111111.11111111**.** Здесь префикс провайдера имеет длину 14 разрядов - 11000001.000101, что можно записать в виде - 193.20.0.0/14. Префикс обычно интерпретируется как номер подсети.

Даже если необходимое клиенту адресное пространство может быть обеспечено предоставлением нескольких сетей стандартного класса, предпочтительным считается вариант IP-адрес/маска, так как в этом случае адреса гарантированно образуют непрерывное пространство. Непрерывность адресного пространства является очень важным свойством, непосредственно влияющим на эффективность маршрутизации.

Рассмотрим еще один пример. Пусть *клиент* собирается связать в сеть 500 компьютеров. Вместо того чтобы выделять ему две сети класса С по 256 узлов каждая, клиенту назначают пул адресов в виде пары 193.20.30.0/23. Эта запись означает, что клиенту выделена сеть неопределенного класса, в которой под нумерацию узлов отведено 9 младших битов, что, как и в случае двух сетей класса С, позволяет адресовать 512 узлов. Преимущество этого варианта с маской перед вариантом с двумя сетями состоит в том, что в первом случае *непрерывность пула адресов гарантирована*.

Назначение адресов в виде IP-адрес/маска корректно лишь в том случае, если поле для адресации узлов, полученное применением маски к IP-адресу, содержит только одни нули. Например, определение пула адресов в виде 193.20.00.0/12 ошибочно, так как в поле номера сети (в 20 младших битах) содержится не нулевое значение 0100.0000 0000.0000 0000. В то же время префикс может оканчиваться нулями, например определение пула 193.20.0.0/25, в котором префикс имеет значение 1100 0001.0001 0100.0000 0000.0, вполне корректно.

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

IPv6 как развитие стека TCP/IP

В начале 90-х годов стек протоколов TCP/IP столкнулся с серьезными проблемами. Именно в это время началось активное промышленное использование Интернета: переход к построению сетей предприятий на основе транспорта Интернета, применение веб-технологии для доступа к корпоративной информации, ведение электронной коммерции через Интернет, внедрение Интернета в индустрию развлечений (распространение видеофильмов, звукозаписей, интерактивные игры).

Все это привело к резкому росту числа узлов сети (в начале 90-х годов новый узел в Интернете появлялся каждые 30 секунд), изменению характера трафика и ужесточению требований, предъявляемых к качеству обслуживания сетью ее пользователей.

Сообщество Интернета, а вслед за ним и весь телекоммуникационный мир, начали решать новые задачи путем создания новых протоколов для стека TCP/IP, таких как протокол резервирования ресурсов (RSVP), защищенный протокол IP (IPSec), протокол коммутации меток (MPLS) и т.п. Однако ведущим специалистам было ясно, что только за счет добавления новых протоколов технологию TCP/IP развивать нельзя - нужно решиться на *модернизацию сердцевины стека*, протокола IP. Некоторые проблемы нельзя было решить без изменения формата IP-пакета и логики обработки полей заголовка IP-пакетов. Наиболее очевидной проблемой такого рода была проблема дефицита IP-адресов, которую невозможно снять, не расширив размер полей адресов источника и приемника.

Критике стала все чаще подвергаться схема масштабирования маршрутизации. Дело в том, что быстрый рост сети вызвал перегрузку маршрутизаторов, которым приходится обрабатывать в своих таблицах маршрутизации информацию о нескольких десятках тысяч номеров сетей, да еще решать некоторые вспомогательные задачи, такие, например, как фрагментация пакетов. Некоторые из предлагаемых решений данной проблемы также требовали внесения изменений в протокол IP.

Наряду с добавлением новых функций непосредственно в протокол IP необходимо было обеспечить его тесное взаимодействие с новыми протоколами - членами стека TCP/IP, что также требовало добавления в заголовок IP новых полей, обработку которых осуществляли бы эти протоколы. Например, для работы RSVP было желательно введение в заголовок IP поля метки потока, а для протокола IPSec - специальных полей для передачи данных, поддерживающих его функции обеспечения безопасности.

В результате сообщество Интернета после достаточно долгого обсуждения решило подвергнуть протокол IP серьезной переработке, выбрав в качестве основных целей модернизации:

- создание масштабируемой схемы адресации;
- сокращение объема работы, выполняемой маршрутизаторами;
- предоставление гарантий качества транспортных услуг;
- обеспечение защиты данных, передаваемых по сети.

Адрес протокола IPv6 состоит из 128 бит, то есть, он в 4 раза длиннее 32-битного IPv4 адреса. Подобно IPv4, в этом адресе можно выделить две части: сеть и хост. То есть, не все биты в адресе имеют одинаковое значение. Часть битов слева (сколько именно зависит от префикса) обозначают сеть, остальные биты справа – идентифицируют устройство внутри сети. Часть, ответственная за хранение информации о хосте называется идентификатор интерфейса (interface id). В отличие от предыдущей версии протокола, в IPv6 не применяются маски подсети, так как они получились бы очень длинными, вместо этого используется префикс. который записывается так же через слеш после адреса. Например, префикс /64 означает, что из 128 бит, первые 64 – это сеть, а оставшаяся часть (в данном случае вторые 64) – это хост. Префикс описывает, сколько бит в адресе используется под хранение информации о сети.

Сам адрес записывают не в десятичном, а в шестнадцатеричном виде - так короче. Адрес разбивается на группы по 16 бит (хекстеты) и каждая группа представляется четырьмя

шестнадцатеричными цифрами. Хекстеты отделяются друг от друга знаком двоеточия. Таким образом, адрес состоит из 8 хекстетов ([8 хекстетов]*[16 бит в хекстете]=[128 бит] - общая длина адреса).

Сокращение IPv6

Пример адреса: 2001:0DB8:AA10:0001:0000:0000:0000:00FB. С таким длинным адресом работать достаточно неудобно, поэтому применяют сокращённую запись.
Для того чтобы сократить данный адрес надо последовательно применить два правила.

Правило 1

В каждом хекстете (группе из 4-х цифр) ведущие нули удаляются. Например, во втором хекстете 0DB0 заменяется на DB0. То есть ноль слева удаляется, ноль справа мы не трогаем. Если хекстет состоит из одних нулей, то он заменяется на один нуль. Таким образом адрес 2001:0DB0:0000:123A:0000:0000:0000:0030 преобразуется в 2001:DB0:0:123A:0:0:0:30. А, например, адрес loopback 0000:0000:0000:0000:0000:0000:0000:0001 заменяется на 0:0:0:0:0:0:0:1.

Правило 2

Это правило применяется только после первого. В адрес выбирается одна самая длинная группа, состоящая из полностью нулевых хекстетов, то есть самая длинная последовательность «:0:0:0:» и заменяется на два двоеточия «::». Эту замену можно произвести только один раз и только с самой длинной последовательностью, так как, если бы мы, например, сделали такую замену в двух местах адреса, то потом нельзя было бы восстановить, сколько именно хекстетов мы заменили в первом и во втором случае. **Важный момент:** нельзя заменять одну группу из :0: на ::, правило два применимо только если есть более одной нулевой группы.
Для примера возьмём адрес из предыдущей замены 2001:DB0:0:123A:0:0:0:30. Самая длинная последовательность из полностью пустых хекстетов – это «:0:0:0:», она начинается сразу после хекстета «123A». Есть ещё последовательность из одного пустого хекстета (между «DB0» и «123A»), но эта – длиннее, так что заменять будем её. Адрес станет совсем небольшим: 2001:DB0:0:123A::30 конечно, длиннее IPv4 адреса, но гораздо короче исходного.

Получение исходного адреса по сокращённой записи

Эта процедура достаточно тривиальна, если мы уже умеем сокращать адреса.
Сначала надо посчитать, сколько хекстетов в адресе осталось. В нашем случае, в адресе 2001:DB0:0:123A::30 осталось 5 хекстетов. Мы знаем, что адрес должен состоять из восьми хекстетов – значит вместо «::» возвращаем три недостающих нулевых, получаем 2001:DB0:0:123A:0:0:0:30. Теперь в каждой группе, где меньше четырёх цифр дописываем слева такое количество нулей, чтобы в группе стало четыре цифры. В результате получим исходный адрес 2001:0DB0:0000:123A:0000:0000:0000:0030.

Примеры

Теперь, чтобы закрепить понимание, приведём несколько примеров сокращения адресов.
Сокращать будем по правилам в два этапа.

- FF80:0000:0000:0000:0123:1234:ABCD:EF12 → FF80:0:0:0:123:1234:ABCD:EF12 → FF80::123:1234:ABCD:EF12
- FF02:0000:0000:0000:0000:0001:FF00:0300 → FF02:0:0:0:0:1:FF00:300 → FF02::1:FF00:300
- 2001:0DB8:0000:1111:0000:0000:0000:0200 → 2001:DB8:0:1111:0:0:0:200 → 2001:DB8:0:1111::200
- 0000:0000:0000:0000:0000:0000:0000:0001 → 0:0:0:0:0:0:0:1 → ::1
- 0000:0000:0000:0000:0000:0000:0000:0000 → 0:0:0:0:0:0:0:0 → ::

Адрес loopback выглядит в сокращённой записи особенно элегантно ::1. Даже если вы не пользуетесь IPv6, но работаете на одной из современных операционных, систем, у вас наверняка установлен этот протокол. Это легко проверить, пропинговав loopback.

При использовании IPv6 адреса в качестве URL, его необходимо заключать в квадратные скобки, при этом, если необходимо указать в URL-е порт, то его следует писать за пределами скобок – http://[2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d]:8080/.

Виды IPv6 адресов

Выделяется несколько типов адресов:

- Глобальный юникаст (*Global unicast*) – это аналог публичных адресов в IPv4. Большая часть всех адресов относятся именно к этому классу. Эти адреса должны быть уникальными в пределах всего интернета, они выдаются IANA региональным регистраторам, те выдают их провайдерам, а провайдеры – выдают клиентам. Диапазон этих адресов – это все адреса, у которых первые три бита равны «001», что означает все адреса, у которых первый хекстет лежит в диапазоне от 2000 до 3FFF. Из этой группы отдельно выделяется сеть 2001:0DB8::/32, которая, согласно спецификации, используется для примеров и документации.
- Локальные адреса (*Link-local*) – адреса, использующиеся для взаимодействия с другими устройствами в той же локальной сети. Отличительной особенностью этих адресов является то, что трафик «с» или «на» эти адреса не маршрутизируется и в принципе не может выйти за пределы той сети, в которой он был создан. Уникальность от этих адресов не требуется – в каждой сети они могут быть одними и теми же. Адреса применяются для разных специальных целей, например, для процедуры обнаружения соседей (аналог ARP в IPv6). Диапазон таких адресов FE80::/10 – что означает все адреса у которых первый хекстет в диапазоне от FE80 до FEBF.
- Мультикастовые адресе (*Multicast*) – адреса, использующийся для мультикастовой рассылки. Все эти адреса находятся в диапазоне FF00::/8, что по-русски означает «Всё что начинается с FF». Надо сказать, что мультикаст в IPv6 выполняет важную роль, так как в нём нет широковещательных пакетов и все рассылки делаются мультикастом. Это очень большая тема, поэтому о мультикастах в IPv6 мы поговорим в одной из следующих статей.
- *Loopback* – специальный адрес ::1. Все пакеты, идущие на него не выходят за пределы устройства, а попадают обратно на уровень IP. Таким образом, этот адрес аналогичен 127.0.0.1 в IPv4. Командой ping ::1 можно проверить, установлен ли на компьютере стек протоколов TCP/IP и IPv6 в частности.
- Неопределённый адрес (*Unspecified address*) – адрес, состоящий из одних нулей. Записывается в сокращённой форме как «::». Такой адрес не может быть назначен

интерфейсу, но может использоваться в некоторых пакетах в качестве адреса отправителя. Например, когда устройство ещё не получило IP адрес с помощью автоматической конфигурации, о ней – тоже в одной из следующих статей.

- Уникальные локальные адреса (*Unique local*) – аналог приватных адресов в IPv4, то есть они могут маршрутизироваться в пределах нашей внутренней сети, но в интернет их анонсировать нельзя. Вообще, IPv6 подразумевает отказ от приватных адресов в том смысле, в котором они использовались до этого. В IPv4 приватные адреса применяются в основном из-за нехватки публичных и только иногда из соображений безопасности. В IPv6 использовать локальные адреса надо только в том случае, если по соображениям безопасности трафик из данной сети и в неё не должен уходить за пределы нашей зоны ответственности. Во всех остальных случаях следует использовать глобальные юникастовые адреса.
- Адреса IPv4, отображенные в IPv6 (*IPv4 embedded*) – это адреса вида ::ffff:xxxx:xxxx, где xxxx:xxxx – это некоторый IPv4 адрес, переведённый в шестнадцатеричный вид. Эти адреса используются для устройств, не поддерживающих IPv6 и обеспечивают способ отображения адресного пространства старой версии протокола в адресное пространство новой.

Выше было сказано, что клиенту, как правило, выдаётся огромная сеть (64 бита префикс), а первые 64 бита – это идентификатор сети. Понятно, однако, что сама эта сеть тоже имеет иерархическую структуру. Как правило, региональный регистратор отдаёт провайдеру сеть с префиксом /48, а провайдер добавляет от себя ещё 16 бит и получает 65536 сетей с префиксом /64, которые затем отдаёт своим клиентам.

Типы вещания в IPv6

В IPv6 существует три вида рассылки:

1. **Unicast** – один источник, один получатель
2. **Multicast** – один источник, несколько получателей
3. **Anycast** – один источник, несколько потенциальных получателей, но отсылается только одному из них.

Как мы видим, здесь нет широковещательной (*Broadcast*) рассылки. Там, где раньше использовалось широковещание, в IPv6 используются мультикастовые адреса. Действительно, зачем ограничивать себя рамками broadcast, когда multicast гораздо гибче – иногда можно отослать сообщение группе хостов, а иногда – всем.

Подробнее о мультикастах

В IPv6 для мультикастовых адресов выделен специальный диапазон FF00::/8. То есть, **все адреса, начинающиеся с FF – мультикастовые**.

Адреса мультикаст бывают двух типов:

1. Назначенные (Assigned multicast) – специальные адреса, назначение которых предопределено.

2. Запрошенные (Solicited multicast) – остальные адреса, которые устройства могут использовать для прикладных задач.

Назначенные адреса

Назначенные адреса – это зарезервированные для определённых групп устройств мультикастовые адреса. Отправляемый на такой адрес пакет будет получен всеми устройствами, входящими в группу. Существует два специальных назначенных мультикастовых адреса:

1. FF02::1 – в эту группу входят все устройства в локальной сети. Таким образом, данный специальный мультикастовый адрес ведёт себя как широковещательный адрес в IPv4. Все устройства обязаны принимать пакеты, отправленные на FF02::1.
2. FF02::2 – в эту группу входят все маршрутизаторы. С помощью данного адреса возможно сделать рассылку по маршрутизаторам, присутствующим в локальной сети. Как только на маршрутизаторе Cisco включается режим маршрутизации для IPv6, он автоматически становится участником этой группы и начинает принимать весь трафик, адресованный на FF02::2.

Запрошенные адреса

Адрес этого типа автоматически появляется, когда на некотором интерфейсе появляется юникастовый адрес. Адрес формируется из сети FF02:0:0:0:0:1:FF00::/104, оствашиеся 24 бита – такие же как у настроенного юникастового адреса. Обратите внимание, что /104 означает, что из данной сети только первые 104 бита берутся для формирования адреса (последние два нуля в записи не участвуют).

Когда некоторое устройство получает пакет, у которого адреса получателя находится в сети FF02:0:0:0:0:1:FF00/104, оно обязано принять этот пакет в том случае, если оставшиеся 24 бита в этом адреса равны последним 24-м битам в юникастовом адресе самого устойства.

Допустим, есть устройство с адресом 2001:0DB8:ABCD:0001:0000:0000:0123:A050. Последние 24 бита (6 шестнадцатеричных цифр) в данном случае – это «23:A050». Значит это устройство обязано принимать так же и трафик, приходящий на муьтикастовый адрес FF02:0:0:0:0:1:FF23:A050.

Такие мультикастовые адреса активно используются в ситуации, когда некоторое устройство хочет узнать MAC адрес своего соседа (аналог ARP в IPv4).

Например, некоторое устройство хочет отправить пакет на локальный адрес FE80::1234:5678. Это локальный адрес (так как начинается с FE80, подробнее о типах адресов в предыдущей статье), значит он находится в нашей локальной сети. Соответственно, чтобы на него что-то отправить, надо узнать MAC-адрес целевого устройства (для формирования Ethernet-фрейма). Хост отправляет на запрошенный (solicited) мультикастовый адрес, FF02:0:0:0:0:1:FF34:5678 пакет. Последние 24 бита «34:5678» были взяты из IPv6-адреса искомого устройства. В случае связки IPv4 и ARP, этот пакет отправлялся бы на адрес 255.255.255.255.

Далее искомое устройство отвечает на этот мультикастовый пакет юникастом, сообщая в ответе свой MAC-адрес.

Возможны ситуации, когда несколько устройств обработают такой пакет (если у них совпадают последние 24 бита адреса), но в этом нет ничего страшного, так как это в любом случае лучше широковещания, когда все устройства обрабатывали запрос.

Автоконфигурация

В IPv6 появился новый механизм автоконфигурации узла. Называется он Stateless Address Autoconfiguration или SLAAC. Используется он для автоматического получения IP адреса и сетевого префикса узлом, без использования DHCPv6 сервера, или совместно с ним.

Действительно, когда мы создаём некоторую сеть, мы прописываем адрес шлюза и префикс этой сети на маршрутизаторе. Этой информации достаточно, чтобы выдавать адреса устройствам. Механизм SLAAC позволяет маршрутизатору назначать устройствам адреса даже если в сети нет DHCPv6.

Маршрутизатор Cisco с рабочим IPv6 интерфейсом рассылает в сеть информацию об этой сети, включающую в себя сетевую часть IP адреса и длину префикса. Кроме того, в этом сообщении содержится адрес шлюза по умолчанию для сети. Сообщение это называется Router Advertisement (RA) и отправляется обычно раз в 200 секунд на мультикастовый адрес FF02::.

Если в сети появилось новое устройство, которому необходим адрес, ему необязательно ждать 200 секунд до ближайшей рассылки, оно может направит запрос маршрутизатору (Router Solicitation или RS) и попросить его выслать настройки немедленно. Запрос маршрутизатору выполняется на адрес FF02::2.

Оба сообщения RA и RS отправляются посредством протокола ICMPv6, с мультикастовым адресом получателя в IP пакете.

Для того чтобы маршрутизатор начал полноценно обслуживать сеть (рассылать в неё RA и отвечать на RS), мало настроить IPv6 адрес на интерфейсе, необходимо так же включить режим маршрутизации для IPv6 сетей, введя команду `ipv6 unicast routing` в режиме глобальной конфигурации.

Существует три способа назначения адреса:

1. Маршрутизатор выдаёт подсеть, префикс и адрес шлюза. Другую информацию устройства не получают.
2. Маршрутизатор выдаёт подсеть, префикс и адрес шлюза, а отдельный DHCPv6 сервер выдаёт дополнительную информацию: опции, маршруты, адреса DNS серверов и другую – по необходимости.
3. Stateless Address Autoconfiguration вообще не используется – устройство не использует RA от маршрутизатора, а обращается к DHCPv6 серверу, который предоставляет всю необходимую информацию, включая адрес, шлюз, префикс, DNS сервера и другую – в зависимости от настроек DHCP.

В случае использования третьего варианта DHCP сервер выдаёт клиенту полный IPv6 адрес – все 128 бит, который назначается на интерфейсе клиента. В случае использования первых двух вариантов, маршрутизатор сообщает клиенту только сеть, в которой он находится, шлюз и префикс. Таким образом, клиенту недостаёт второй половины IP адреса (идентификатора интерфейса). Напомню, что адрес состоит из 128 бит, а маршрутизатор выдаёт максимум, только первые 64. Оставшиеся 64 бита, где должна находиться информация о хосте, должны быть заполнены самим устройством, маршрутизатору не важно, что именно устройство туда поместит, важно, чтобы первые 64 бита (сеть) были правильными. Для генерации правой половины IP адреса используется алгоритм EUI-64 или вообще генерируется случайный набор цифр.

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 назначить клиенту не только IP-адрес, но и другие параметры стека TCP/IP, необходимые для его эффективной работы, например маску и IP-адрес маршрутизатора, предлагаемые 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, гарантируя от дублирования адресов за счет централизованного управления их распределением.

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 DHCP-сервер 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 попрежнему, 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.
Покажем преимущества, которые дает динамическое распределение пула адресов, на примере. Пусть в некоторой организации 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 С 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 коннекторами для подключения компьютеров. Но возникает другая проблема —кто и как будет конфигурировать компьютеры, состав которых постоянно меняется?
Существуют 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 за пускается 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 у 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 (такой вариант соответствует сети 1).
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 сети 2 DHCP-серверу сети 3. 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 со стояние.
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. Это 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 поля 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.

Основы маршрутизации:

IP-пакет

В каждой очередной сети, лежащей на пути перемещения пакета, протокол IP обращается к средствам транспортировки этой сети, чтобы с их помощью передать пакет на маршрутизатор, ведущий к следующей сети, или непосредственно на узел-получатель. *Поддержание интерфейса с нижележащими технологиями* подсетей является одной из важнейших функций протокола IP. В эти функции входит также *поддержание интерфейса с протоколами вышележащего транспортного уровня*, в частности с протоколом TCP, который решает все вопросы обеспечения надежной доставки данных по составной сети в стеке TCP/IP.

Протокол IP относится к протоколам *без установления соединений*, он поддерживает обработку каждого IP-пакета как независимой единицы обмена, не связанной с другими пакетами. В протоколе IP нет механизмов, обычно применяемых для обеспечения достоверности конечных данных. Если во время продвижения пакета происходит какая-либо ошибка, то протокол IP по своей инициативе ничего не предпринимает для исправления этой ошибки. Например, если на промежуточном маршрутизаторе пакет был отброшен из-за ошибки по контрольной сумме, то модуль IP не пытается заново послать потерянный пакет. Другими словами, протокол IP реализует политику доставки «по возможности».

Имеется прямая связь между количеством полей заголовка пакета и функциональной сложностью протокола, который работает с этим заголовком. Чем проще заголовок - тем проще соответствующий протокол. Большая часть действий протокола связана с обработкой той служебной информации, которая переносится в полях заголовка пакета. Изучая назначение каждого поля заголовка IP-пакета, мы не только получаем формальные знания о структуре пакета, но и знакомимся с основными функциями протокола IP.

Поле номера версии занимает 4 бита и идентифицирует версию протокола IP. Сейчас повсеместно используется версия 4 (IPv4), хотя все чаще встречается и новая версия (IPv6).

Значение **длины заголовка** IP-пакета также занимает 4 бита и измеряется в 32-битных словах. Обычно заголовок имеет длину в 20 байт (пять 32-битных слов), но при добавлении некоторой служебной информации это значение может быть увеличено за счет дополни тельных байтов в поле параметров. Наибольшая длина заголовка составляет 60 байт.

Поле **типа сервиса** (Type of Service, ToS) имеет и другое, более современное название - **байт дифференцированного обслуживания,** или **DS-байт.** Этим двум названиям соответствуют два варианта интерпретации этого поля. В обоих случаях данное поле служит одной цели —хранению признаков, которые отражают требования к качеству обслуживания пакета. В прежнем варианте первые три бита содержат значение **приоритета** пакета: от самого низкого —0 до самого высокого - 7. Маршрутизаторы и компьютеры могут принимать во внимание приоритет пакета и обрабатывать более важные пакеты в первую очередь. Следующие три бита поля ToS определяют **критерий выбора маршрута.** Если бит D (Delay - задержка) установлен в 1, то маршрут должен выбираться для минимизации задержки доставки данного пакета, установленный бит T (Throughput - пропускная способность) - для максимизации пропускной способности, а бит R (Reliability - надежность) - для максимизации надежности доставки. Оставшиеся два бита имеют нулевое значение.

Стандарты дифференцированного обслуживания, принятые в конце 90-х годов, дали новое название этому полю и переопределили назначение его битов. В DS-байте также используются только старшие 6 бит, а два младших бита остаются в качестве резерва. Назначение битов DS-байта рассмотрено в разделе «Поддержка QoS в машрутизаторах» главы 17.

Поле **общей длины** занимает 2 байта и характеризует общую длину пакета с учетом заголовка и поля данных. Максимальная длина пакета ограничена разрядностью поля, определяющего эту величину, и составляет 65 535 байт, однако в большинстве компьютеров и сетей столь большие пакеты не используются. При передаче по сетям различного типа длина пакета выбирается с учетом максимальной длины пакета протокола нижнего уровня, несущего IP-пакеты. Если это кадры Ethernet, то выбираются пакеты с максимальной длиной 1500 байт, умещающиеся в поле данных кадра Ethernet. В стандартах TCP/IP предусматривается, что все хосты должны быть готовы принимать пакеты длиной вплоть до 576 байт (независимо от того, приходят ли они целиком или фрагментами).

Идентификатор пакета занимает 2 байта и используется для распознавания пакетов, образовавшихся путем деления на части (фрагментации) исходного пакета. Все части (фрагменты) одного пакета должны иметь одинаковое значение этого поля.

Флаги занимают 3 бита и содержат признаки, связанные с фрагментацией. Установленный в 1 бит DF (Do not Fragment - не фрагментировать) запрещает маршрутизатору фрагментировать данный пакет, а установленный в 1 бит MF (More Fragments - больше фрагментов) говорит о том, что данный пакет является промежуточным (не последним) фрагментом. Оставшийся бит зарезервирован.

Поле **смещения фрагмента** занимает 13 бит и задает смещение в байтах поля данных этого фрагмента относительно начала поля данных исходного (нефрагментированного) пакета.

Используется при сборке/разборке фрагментов пакетов. Смещение должно быть кратно 8 байт.

Поле **времени жизни** (Time To Live, TTL) занимает один байт и используется для задания предельного срока, в течение которого пакет может перемещаться по сети. Время жизни пакета измеряется в секундах и задается источником. По истечении каждой секунды пребывания на каждом из маршрутизаторов, через которые проходит пакет во время своего «путешествия» по сети, из его текущего времени жизни вычитается единица; единица вычитается и в том случае, если время пребывания было меньше секунды. Поскольку современные маршрутизаторы редко обрабатывают пакет дольше чем за одну секунду, то время жизни можно интерпретировать как максимальное число транзитных узлов, которые разрешено пройти пакету. Если значение поля времени жизни становится нулевым до того, как пакет достигает получателя, пакет уничтожается. Таким образом, время жизни является своего рода часовым механизмом самоуничтожения пакета.

Поле **протокола верхнего уровня** занимает один байт и содержит идентификатор, указывающий, какому протоколу верхнего уровня принадлежит информация, размещенная в поле данных пакета. Значения идентификаторов для разных протоколов приводятся в документе RFC 1700, доступном по адресу http://www.iana.org. Например, 6 означает, что в пакете находится сообщение протокола TCP, 17 - протокола UDP, 1 - протокола ICMP.

Контрольная сумма заголовка занимает 2 байта (16 бит) и рассчитывается только по заголовку. Поскольку некоторые поля заголовка меняют свое значение в процессе передачи пакета по сети (например, поле времени жизни), контрольная сумма проверяется и повторно рассчитывается на каждом маршрутизаторе и конечном узле как дополнение к сумме всех 16-битных слов заголовка. При вычислении контрольной суммы значение самого поля контрольной суммы устанавливается в нуль. Если контрольная сумма неверна, то пакет отбрасывается, как только обнаруживается ошибка.

Поля **IP-адресов источника** и **приемника** имеют одинаковую длину - 32 бита.

Поле **параметров** является не обязательным и используется обычно только при отладке сети. Это поле состоит из нескольких подполей одного из восьми предопределенных типов. В этих подполях можно указывать точный маршрут, по которому маршрутизаторы должны направлять данный пакет (это называется **маршрутизацией от источника),** регистрировать проходимые пакетом маршрутизаторы или помещать данные системы безопасности и временные отметки. Так как число подполей в поле параметров может быть произвольным, то в конце заголовка должно быть добавлено несколько нулевых байтов для **выравнивания** заголовка пакета по 32-битной границе.

Далее приведена распечатка значений полей заголовка одного из реальных 1P-пакетов, захваченных в сети Ethernet средствами анализатора протоколов сетевого монитора (Network Monitor, NM) компании Microsoft. В данной распечатке NM в скобках дает шестнадцатеричные значения полей, кроме того, программа иногда представляет числовые коды полей в виде, более удобном для чтения. Например, дружественный программный интерфейс NM интерпретирует код 6 в *поле протокола верхнего уровня*, помещая туда название соответствующего протокола - TCP (см. строку, выделенную полужирным шрифтом).

```
IP: Version = 4 (0x4)
IP: Header Length = 20 (0x14)
IP: Service Type = 0 (0x0)
IP: Precedence = Routine
IP: ...0--- = Normal Delay
IP: ....0... = Normal Throughput IP: .... 0.. = Normal Reliability IP: Total Length = 54 (0x36)
IP: Identification = 31746 (0x7C02)
```

```
IP: Flags Summary = 2 (0x2)
IP: ..... 0 = Last fragment in datagram
IP: ..... 1. = Cannot fragment datagram
IP: Fragment Offset = 0 (0x0) bytes
IP: Time to Live = 128 (0x80)
IP: Protocol = TCP - Transmission Control
IP: Checksum = 0xEB86
IP: Source Address = 194.85.135.75
IP: Destination Address = 194.85.135.66
IP: Data: Number of data bytes remaining = 34 (0x0022)
```

Маршрутизация IP

Маршрутизация IP (процесс перенаправления пакетов IP) обеспечивает доставку пакетов через все сети TCP/IP с устройства, создавшего пакет IP, на устройство его получателя. Другими словами, маршрутизация IP доставляет пакеты IP с хоста отправителя на хост получателя.

Полный процесс сквозной маршрутизации использует логику сетевого уровня на хостах и маршрутизаторах. Для создания и перенаправления пакета IP на стандартный шлюз хоста (стандартный маршрутизатор) передающий хост использует концепции уровня 3. Когда, принимая решение о перенаправлении пакета IP, маршрутизатор сравнивает адрес получателя в пакете с таковым в таблице маршрутизации, также используется логика уровня 3.

Процесс маршрутизации полагается также на физические свойства каждого канала связи. Маршрутизация IP использует последовательные каналы связи, локальные сети Ethernet, беспроводные локальные сети и много других сетей, реализующих стандарты физического уровня и канал связи. Эти низкоуровневые устройства и протоколы перемещают пакеты IP по сети TCP/IP, инкапсулируя и передавая пакеты во фреймах канального уровня.

Процесс маршрутизации IPv4

Процесс маршрутизации начинается с хоста, создающего пакет IP. Сначала хост решает вопрос: не принадлежит ли IP-адрес получателя этого нового пакета локальной подсети? Для определения диапазона адресов в локальной подсети хост использует собственный IP-адрес и маску. На основании собственных выводов о диапазоне адресов локальной подсети хост действует следующим образом.

Этапы перенаправления пакетов IP хостом:

Этап 1

Если получатель локальный, передача осуществляется непосредственно:

A. MAC-адрес хоста получателя определяется при помощи уже существующей записи таблицы протокола преобразования адресов (ARP) или сообщения ARP, позволяющего изучить эту информацию.

B. Пакет IP инкапсулируется во фрейм канала связи с адресом канала связи хоста получателя (destination host)

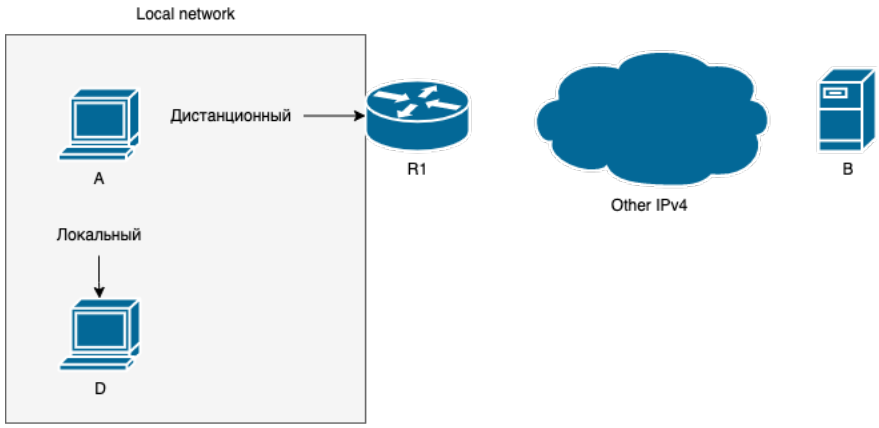
Этап 2

Если получатель не является локальным, то передача осуществляется на стандартный шлюз:

A. MAC-адрес стандартного шлюза определяется при помощи уже существующей записи таблицы ARP или сообщения ARP, позволяющего изучить эту информацию

B. Пакет IP инкапсулируется во фрейм канала связи с адресом канала связи стандартного шлюза (default gateway)

Хост A на рисунке посылает пакет локальному хосту D непосредственно. Но имя хоста B, расположенного с другой стороны маршрутизатора, а следовательно, в другой подсети, хост A посылает пакет на свой стандартный маршрутизатор (RI). (Термины стандартный шлюз (default gateway) и стандартный маршрутизатор (default router) - синонимы.)



У маршрутизаторов немного больше работы при маршрутизации по сравнению с хостами. В то время как логика хоста начинается с пакета IP, находящегося в памяти, маршрутизатору, прежде чем дойти до того положения, необходимо проделать некоторую работу. Ниже приведено пять этапов логики маршрутизации, причем на первых двух этапах осуществляется только получение фрейма и извлечение пакета IP перед принятием решения об адресе получателя пакета на этапе 3.

Этапы перенаправления пакетов IP маршрутизатором

Этап 1

Для каждого полученного фрейма канала связи принимается решение, обрабатывать его или нет. Обрабатывается фрейм так:

A. Проверка фрейма на ошибки (по полю контрольной суммы фрейма (FCS) в концевике канала связи).

B. Адрес канала связи получателя фрейма - это адрес маршрутизатора (или соответствующий много адресатный или широковещательный адрес)

Этап 2

Перед решением об обработке фрейма на этапе 1 он извлекается из фрейма канала связи

Этап 3

Принимается решение о маршрутизации. Для этого по IP-адресу получателя пакета осуществляется поиск соответствующего элемента таблицы маршрутизации, содержащего маршрут к получателю. Этот маршрут идентифицирует исходящий интерфейс маршрутизатора, а возможно, и следующий транзитный маршрутизатор

Этап 4

Помещает (инкапсулирует) пакет во фрейм канала связи, соответствующего и сходящему интерфейсу. По мере необходимости для поиска MAC-адреса следующего устройства используется протокол ARP

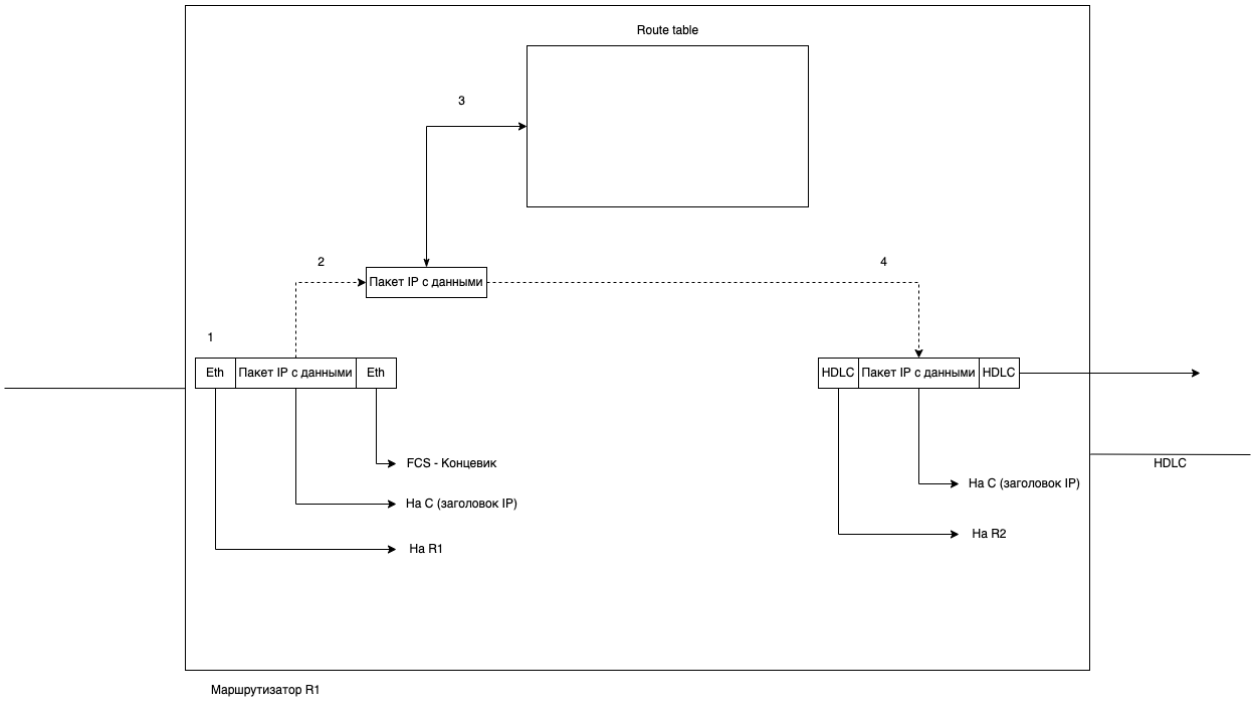
Этап 5

Фрейм передается на исходящий интерфейс, указанный в соответствующем маршруте IP

Этапы процесса маршрутизации насчитывают много подробностей, но иногда его можно рассматривать упрощенно. Например, отбросив некоторые детали, этапы этого процесса можно пересказать следующим образом:

Маршрутизатор получает фрейм, извлекает из него пакет, решает, куда его перенаправить, помещает пакет в другой фрейм и посылает его.

На рисунке показан пакет, поступающий слева на входной интерфейс Ethernet маршрутизатора с IP-адресом хоста получателя C. Пакет поступает инкапсулированным во фрейм Ethernet (с заголовком и концевиком).



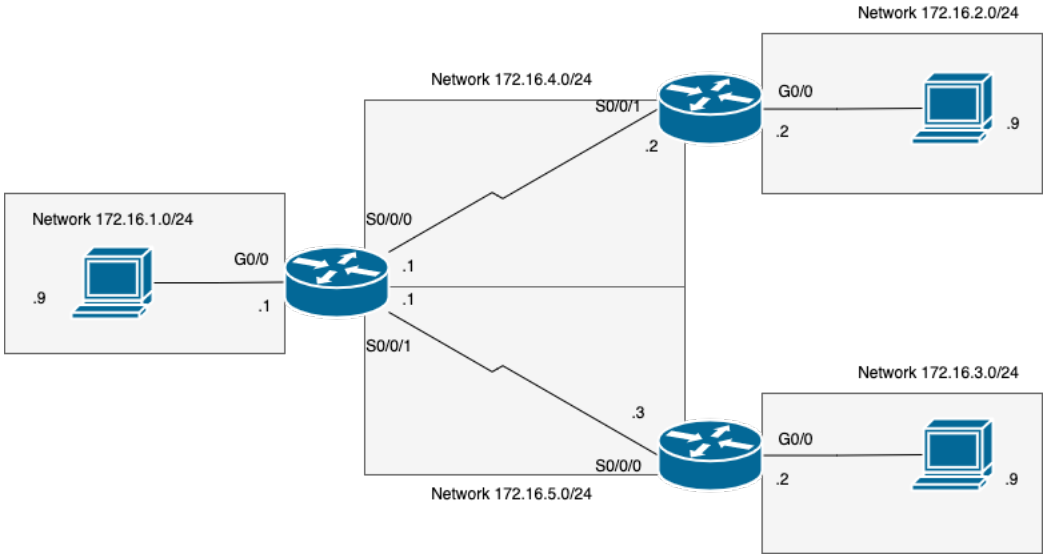
Маршрутизатор R1 обрабатывает фрейм и пакет, как показано цифрами на рисунке в соответствии с процессом из пяти этапов.

1. Маршрутизатор R1 отмечает, что полученный фрейм Ethernet прошел проверку FCS и что получатель Ethernet имеет MAC-адрес маршрутизатора RI, поэтому маршрутизатору RI предстоит обрабатывать фрейм.
2. Маршрутизатор Rl извлекает пакет IP из заголовка и концевика фрейма Ethernet.
3. Маршрутизатор Rl ищет IP-адрес получателя пакета IP в таблице маршрутизации IP.
4. Маршрутизатор Rl инкапсулирует пакет IP в новый фрейм канала связи (в данном случае в заголовок и концевик протокола HDLC).
5. Маршрутизатор Rl передает пакет IP в новом фрейме HDLC через последовательный канал связи справа.

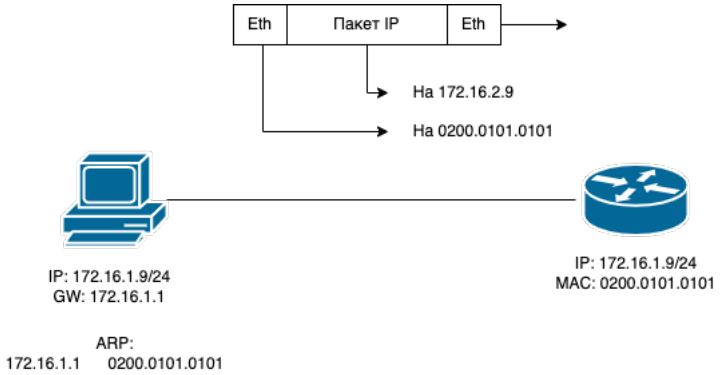
Протокол HDLC (High-Level Data Link Control) – протокол второго уровня модели OSI, разработанный организацией ISO. Этот протокол обеспечивает передачу данных между устройствами в режиме точка-точка или точка-многоточка.

Пример маршрутизации IP

Далее рассматриваются этапы маршрутизации через несколько устройств. В данном случае хост A (172.16.1.9) посылает пакет хосту B (172.16.2.9), используя логику маршрутизации хоста. Затем маршрутизатор Rl перенаправляет пакет согласно логике из пяти этапов.

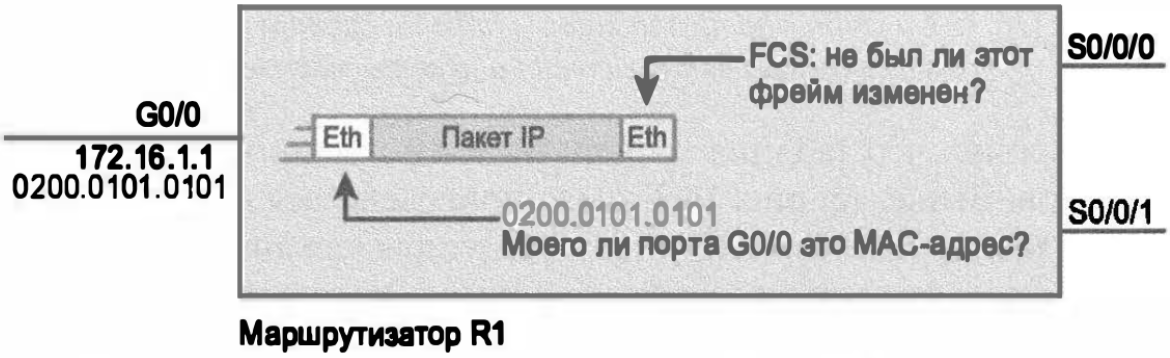


Типичная схема IP-адресации для сети 1 Pv4 с типичными сокращениями адресов. Если отображать полные IP-адреса для каждого интерфейса маршрутизатора, схема окажется слишком загроможденной. По возможности на рисунках обычно указывают подсеть, затем последний (или два) октет IP-адресов, - этого вполне достаточно для идентификации IP-адреса без излишеств. Предположим, например, что хает использует IP-адрес 172.16.1.9 из подсети 172.16.1.0/24 (в которой все адреса начинаются с 172.16.1), поэтому около пиктограммы хоста А изображена цифра ".9". Вот другой пример: маршрутизатор Rl использует адрес 172.16.1.1 на своем интерфейсе LAN, адрес 172.16.4.1 на последовательном интерфейсе и адрес 172.16.5.1 на еще одном последовательном интерфейсе.
Теперь рассмотрим пример с хостом А (172.16.1.9), посылающим пакет хосту В (172.16.2.9). Хост перенаправляет пакет IP на стандартный маршрутизатор (шлюз)
В этом примере хост А использует некое приложение, передающее данные хосту В (172.16.2.9). После формирования хостом А пакета IP в памяти логика хоста А сводится к следующему.
• Мой IP-адрес/маска- 172.16.1.9/24, следовательно, моя локальная подсеть содержит номера 172.16.1.0-172.16.1.255 (включая идентификаторы и широковещательные адреса подсети).
• Пошлю пакет на мой стандартный шлюз по адресу 172.16.1.1.
• Адрес получателя, 172.16.2.9, явно находится не в моей локальной подсети.
• Чтобы послать пакет, инкапсулирую его во фрейме Ethernet. MAC-адрес получателя будет принадлежать интерфейсу G0/0 маршрутизатора Rl (стандартный шлюз хоста А).
IP-адрес и MAC-адрес получателя во фрейме и пакете, посланном хостом А в данном случае.

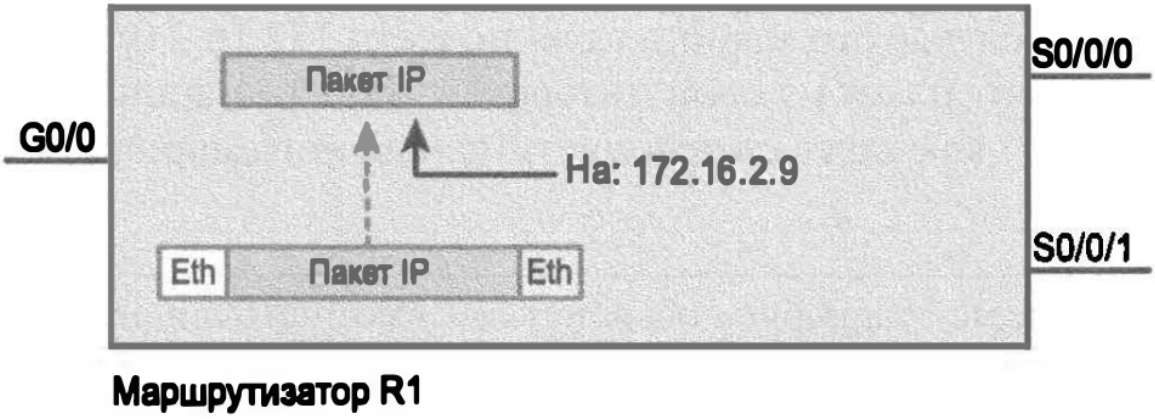


Обратите внимание на то, что каналы Ethernet LAN представлены на рисунке как простые линии, но они могут включать любые устройства, обсуждавшиеся в части 1 1. Канал LAN может быть просто кабелем между хостом А и маршрутизатором RI, или это может быть сотня коммутаторов LAN, объединенных в огромную территориальную сеть. Независимо от этого, хост А и маршрутизатор R l находятся в той же сети VLAN, и локальная сеть Ethernet доставляет фреймы Ethernet на интерфейс G0/0 маршрутизатора R1.
1-й этап маршрутизации: решение об обработке входящих фреймов
Маршрутизаторы получают много фреймов на интерфейсах, в частности на интерфейсах LAN. Но маршрутизатор может и должен игнорировать некоторые из этих фреймов. Поэтому первый этап процесса маршрутизации начинается с решения о том, должен ли маршрутизатор обработать фрейм или отбросить его.
Сначала маршрутизатор осуществляет простую, но очень важную проверку (этап 1 А процесса), - он должен игнорировать все фреймы, переданные с ошибками. Для проверки фрейма на ошибки передачи маршрутизатор использует поле FCS заголовка канала связи. (Маршрутизатор не предпринимает попыток восстановления после ошибок; т.е. не запрашивает повторную передачу данных.)
Маршрутизатор проверяет также адрес канала связи получателя (этап 1 В), чтобы выяснить, предназначен ли фрейм для маршрутизатора. Например, фреймы, посланные на

одноадресатный MAC-адрес интерфейса маршрутизатора, однозначно предназначались ему. Но маршрутизатор вполне может получить фрейм, посланный на некий другой одноадресатный MAC-адрес. Такой фрейм следует игнорировать.
Маршрутизаторы получат одноадресатные фреймы, посланные на другие устройства сети VLAN, благодаря принципу работы коммутаторов LAN. Помните, коммутаторы LAN рассылают одноадресатные фреймы с неизвестным получателем - это фреймы, для которых коммутатор не нашел MAC-адрес получателя в таблице MAC-адресов. Иногда маршрутизаторы получают фреймы, предназначенные для некоего другого устройства, причем с MAC-адресом другого устройства. Такие фреймы маршрутизаторы должны игнорировать.
В этом примере хост А посылает фрейм на MAC-адрес маршрутизатора R l . Таким образом, после получения этого фрейма и проверки его FCS, подтверждающей отсутствие ошибки, маршрутизатор R 1 устанавливает, что фрейм предназначен для MAC-адреса маршрутизатора Rl (в данном случае 0200.0101.0101). Поскольку все проверки пройдены, маршрутизатор Rl решает обработать фрейм, как показано на рис. 16.5. (Обратите внимание на большой прямоугольник на рисунке, он представляет внутреннюю организацию маршрутизатора R1.)

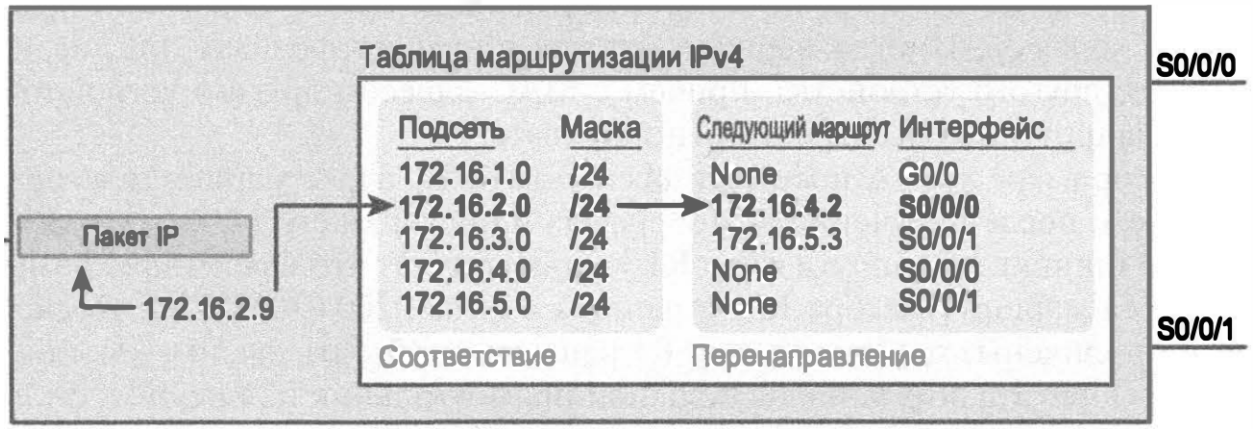


2-й этап маршрутизации: извлечение пакета IP
Выяснив, что полученный фрейм следует обработать (этап 1), маршрутизатор предпринимает следующий шаг - извлекает пакет. В памяти маршрутизатора не нужен ни заголовок, ни концевик канала связи первоначального фрейма, поэтому маршрутизатор удаляет их, оставляя только пакет IP, как показано на рис. 1 6.6. Обратите внимание, что IP-адрес получателя остается неизменным (172.16.2.9).



3-й этап маршрутизации: выбор направления перенаправления пакета
Второй этап маршрутизации несложен, в отличие от этапа 3. Теперь маршрутизатор должен выбрать направление перенаправления пакетов. Для этого используется таблица маршрутизации IP маршрутизатора и логика соответствия при поиске адреса получателя пакета в таблице.

Таблица маршрутизации IP содержит несколько записей маршрутов. Каждая запись маршрута содержит несколько фактов, которые в свою очередь могут быть сгруппированы. Часть записи используется для поиска соответствия адресу получателя пакета, в то время как остальная часть записи содержит инструкцию по перенаправлению, т.е. куда послать пакет.



Маршрутизатор R1

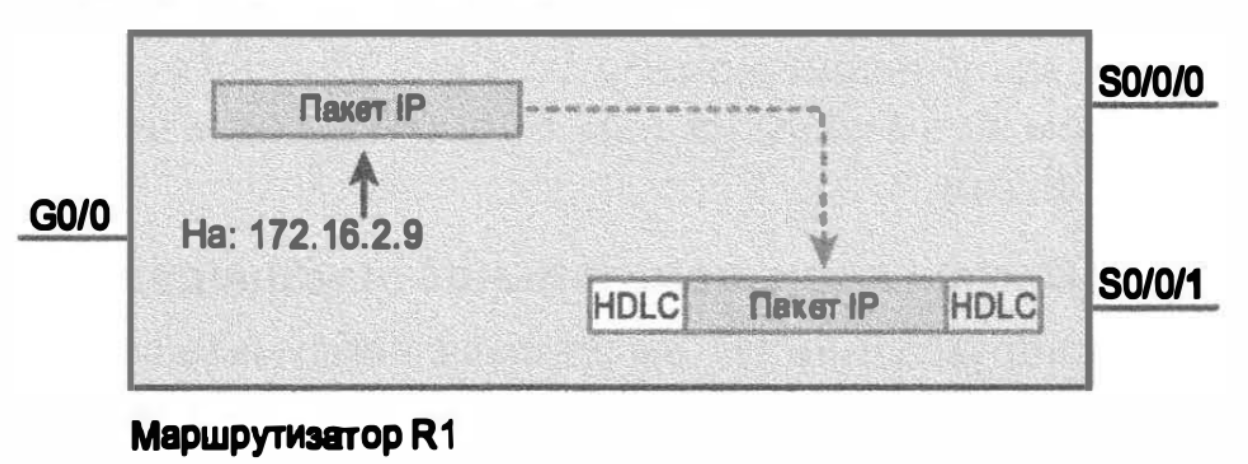
Обратите внимание, что таблица маршрутизации в данном случае содержит пять записей маршрутов. Таким образом, у маршрутизатора Rl есть маршрут для каждой из этих пяти подсетей.

Теперь рассмотрим ту часть записей маршрутов, которую маршрутизатор Rl будет использовать для поиска соответствия пакету. Для полного определения подсети каждая запись маршрута содержит идентификатор и маску подсети. Маршрутизатор ищет соответствие IP-адреса получателя пакета (172.16.2.9) в таблице маршрутизации, сравнивая его с диапазоном адресов, определенных каждой подсетью. Точнее, маршрутизатор просматривает информацию о подсети и маске, для которой достаточно применить несколько математических действий, чтобы выяснить, в какой из этих подсетей располагается адрес 172.16.2.9 (в подсети 172.16.2.0/24). И наконец, обратимся к правой части рисунка - к инструкциям перенаправления для этих пяти маршрутов. После того как маршрутизатор найдет соответствующий маршрут, он использует информацию о перенаправлении, чтобы узнать, куда послать пакет далее. В данном случае это маршрут для подсети 172.16.2.0/24, поэтому маршрутизатор Rl перенаправит пакет на свой интерфейс S0/0/0, маршрутизатору R2, указав

4-й этап маршрутизации: инкапсуляция пакета в новый фрейм

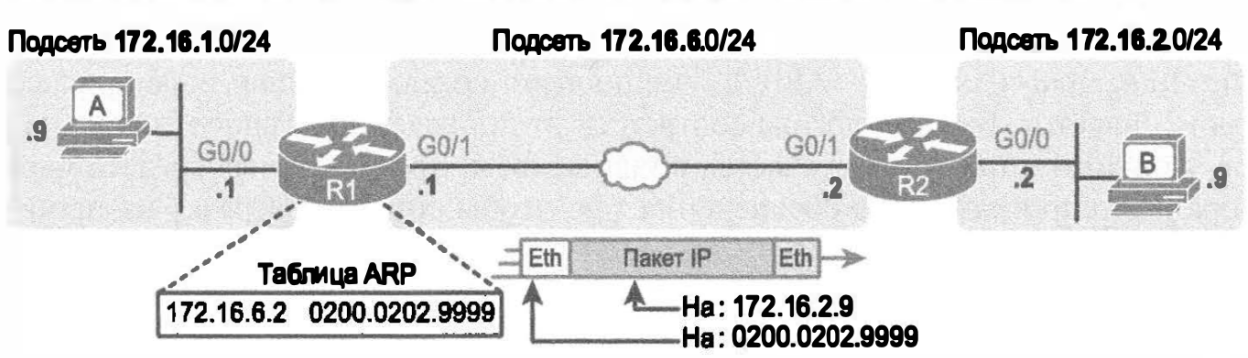
Теперь маршрутизатор знает, куда перенаправить пакет. Но маршрутизаторы не могут перенаправить пакет без оболочки из заголовка и концевика канала связи (инкапсуляция).

Инкапсуляция пакетов для последовательных каналов связи не требует особого размышления из-за простоты протоколов PPP и HDLC. Как упоминалось в главе 3, поскольку последовательные каналы соединяют только два устройства (отправителя и получателя), адресация канала связи не имеет значения. В данном примере маршрутизатор Rl перенаправляет пакет через интерфейс S0/0/0, поместив его во фрейм HDLC.



Маршрутизатор R1

Отметим, что при некоторых других типах каналов связи у маршрутизатора будет побольше работы на этом этапе маршрутизации. Например, иногда маршрутизатор перенаправляет пакеты на интерфейс Ethernet. Чтобы инкапсулировать пакет IP, маршрутизатор должен создать заголовок Ethernet, включающий правильное значение MAC-адреса получателя.
Рассмотрим, например, другую типовую сеть: с каналом связи Ethernet WAN между маршрутизаторами RI и R2. Маршрутизатор RI выбирает маршрут, указывающий перенаправить пакет на интерфейс Ethernet 00/1 для маршрутизатора R2 (172.16.6.2). Для этого маршрутизатор Rl должен поместить в заголовок MAC-адрес маршрутизатора R2. Чтобы сделать это, он использует информацию из таблицы ARP, как показано на рис. 16.9. Если у маршрутизатора Rl нет в таблице ARP записи для адреса 172.16.6.2, то он использует протокол ARP для изучения этого MAC-адреса.



5-й этап маршрутизации: передача фрейма
После завершения подготовки фрейма маршрутизатору остается только передать его. Возможно, маршрутизатору придется подождать, особенно если другие фреймы уже ждут своей очереди для отправки через интерфейс.

Потенциальные проблемы производительности маршрутизации

Изучая процесс маршрутизации IP, имеет смысл обдумать все его частности, обсуждаемые ниже. Таким образом, маршрутизаторы тратят некоторое время на выполнение действий по перенаправлению отдельного пакета IP. Фактически даже очень медленные маршрутизаторы должны перенаправлять десятки тысяч пакетов в секунду; поэтому они не могут нести больших затрат на обработку каждого пакета.
Процесс поиска в таблице маршрутизации соответствия адресу получателя пакета IP фактически может занять много процессорного времени. В примере на рис. 16.7 приведено только пять маршрутов, но в корпоративных сетях обычно есть тысячи маршрутов IP , а у маршрутизаторов ядра Интернета их сотни тысяч. Теперь подумайте о процессоре маршрутизатора, ведь он должен осуществлять поиск в списке из 100000 записей для каждого пакета, а их следует перенаправлять сотни тысяч в секунду! А что если маршрутизатору пришлось бы еще вычислять каждый раз подсети, диапазоны адресов в

каждой подсети для каждого маршрута? Эти вычисления отняли бы слишком много мощности процессора.

За последние годы компания Cisco выработала несколько способов оптимизации внутреннего процесса перенаправления пакетов. Некоторые из них связаны со специфической моделью или серией маршрутизаторов. Коммутаторы уровня 3 осуществляют перенаправление в специализированных интегральных микросхемах (Application Specific Integrated Circuits- ASIC), специально созданных для перенаправления фреймов и пакетов. Базовая логика соответствует приведенному ранее списку из пяти этапов, а оптимизация осуществляется в зависимости от аппаратных средств маршрутизатора и его программного обеспечения так, чтобы снизить нагрузку на процессор и сократить дополнительные затраты на перенаправление пакетов IP.