**Адресация IPv4**

Прежде, чем переходить к рассмотрению IP-адресации не­обходимо вкратце разобраться, как хосты и маршрутизаторы объеди­няются в сеть. Как правило, хост обладает всего одним каналом, кото­рый связывает его с сетью. Когда IP-адресу хоста требуется послать дейтаграмму, передача осуществляется именно через этот канал. Точка сопряжения между хостом и физическим каналом называется **интер­фейсом.** Теперь давайте рассмотрим маршрутизатор и его интерфейсы.

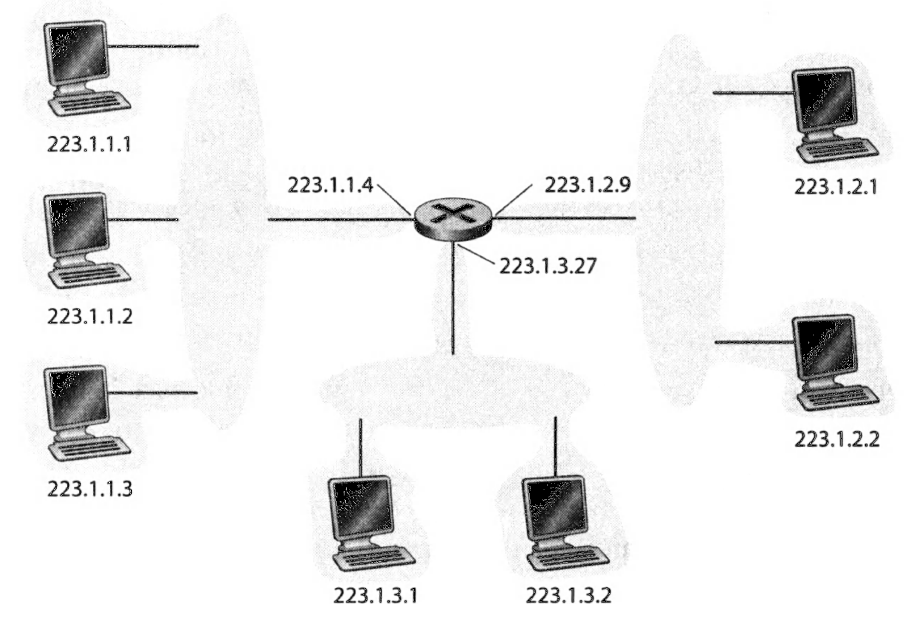
Поскольку задача маршрутизатора — получить дейтаграмму с одного канала и переправить ее по другому, маршрутизатор обязательно свя­зан с двумя или более каналами. Точка сопряжения маршрутизатора с любым из его каналов также называется «интерфейс». Соответствен­но, у маршрутизатора несколько интерфейсов, по одному на каждый входящий в него канал. Поскольку все хосты и маршрутизаторы спо­собны посылать и получать IP-дейтаграммы, IP-протокол требует, чтобы у всех интерфейсов каждого хоста и маршрутизатора были соб­ственные IP-адреса. Таким образом, технически IP-адрес ассоциирован с интерфейсом, который может быть расположен на сетевой карте, а не с хостом или маршрутизатором, содержащим этот интерфейс.

Длина каждого IP-адреса составляет 32 бита (то есть 4 байта). Это означает, что всего могут существовать 232 разнообразных IP-адресов. Аппроксимируя величину 210 к 103, легко вычислить, что точное коли­чество IP-адресов близко к 4 миллиардам. Как правило, эти адреса за­писываются в так называемом **точечно-десятичном формате,** где каж­дый байт адреса представляется в десятичном виде и отделяется точкой от других байтов. Рассмотрим, к примеру, IP-адрес 193.32.216.9. В нем 193 — это десятичный эквивалент первых 8 бит в адресе; 32 — деся­тичный эквивалент вторых 8 бит в адресе и т. д. Следовательно, адрес 193.32.216.9 будет иметь в двоичной системе следующий вид:

11000001 00100000 11011000 00001001

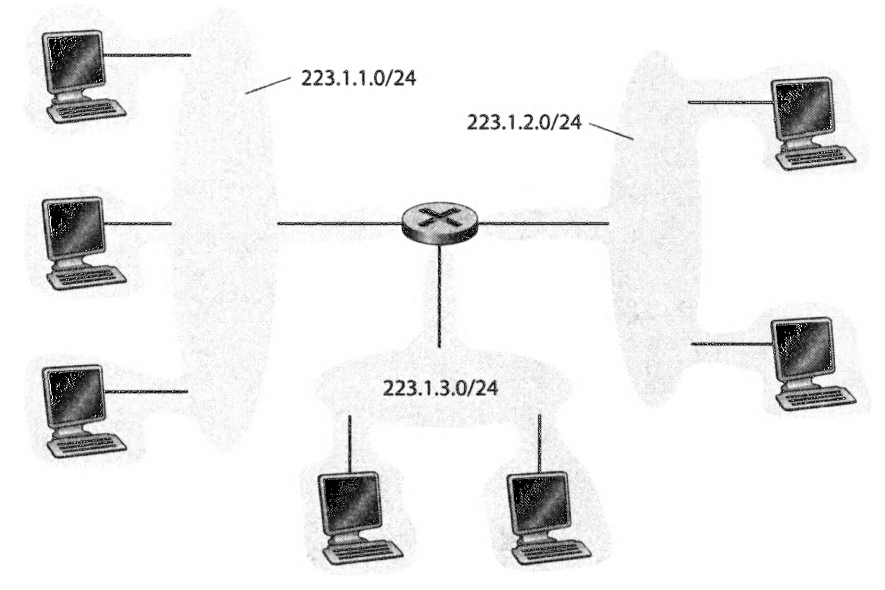
Каждый интерфейс на каждом хосте и маршрутизаторе во всем гло­бальном Интернете должен иметь уникальный IP-адрес (за исключе­нием интерфейсов, расположенных за NAT). Адрес не может быть выбран произвольно. Часть IP-адреса будет определяться той подсетью, к которой подключен интерфейс.

На рис. 4.15 предоставлен пример IP-адресации и расположения ин­терфейсов. Здесь один маршрутизатор с тремя интерфейсами исполь­зуется для взаимного соединения семи хостов. Обратите внимание на то, какие IP-адреса присвоены интерфейсам хостов и маршрутизатора, а именно на несколько вещей. Во-первых, у трех хостов, расположен­ных в верхней левой части рис. 4.15, и у интерфейса маршрутизатора, к которому они подключены, IP-адреса имеют вид 223.1.1.ххх. Кроме того, четыре интерфейса соединены друг с другом сетью, *не содержащей маршрутизаторов.* Эта система могла бы быть связана по локальной сети Ethernet, в случае чего взаимодействие между интерфейсами обес печивалось через Ethernet-коммутатор, либо через беспроводную точку доступа.

**Рис. 4.15. Адреса интерфейсов и подсети**

В контексте IP-протокола принято говорить, что подобная сеть, соеди­няющая три интерфейса хост-машин и один интерфейс маршрутизатора, представляет собой **подсеть.** В технической литературе по Интернету подсеть также иногда называется *IP-сеть* или просто *сеть.* В ходе IP-адресации данной подсети присваивается адрес 223.1.1.0/24. Часть адреса /24, иногда называемая **маской подсети,** указывает, что 24 крайних левых разряда 32-разрядного значения определяют адрес подсети. Соответствен­но, подсеть 223.1.1.0/24 состоит из трех интерфейсов хостов (223.1.1.1, 223.1.1.2 и 223.1.1.3) и одного интерфейса маршрутизатора (223.1.1.4). Лю­бые дополнительные хосты, подключаемые к подсети 223.1.1.0/24, *обяза­тельно должны* будут иметь адреса вида 223.1.1.ххх. На рис. 4.15 мы видим еще две подсети: сеть 223.1.2.0/24 и подсеть 223.1.3.0/24. На рис. 4.16 под­робнее показаны три IP-подсети, присутствующие на рис. 4.15.

IP-определение подсети не ограничено сегментами Ethernet, соеди­няющими множество хостов с интерфейсом маршрутизатора. Чтобы составить впечатление об этой проблеме, рассмотрим рис. 4.17. Здесь показаны три маршрутизатора, все они соединены друг с другом по двух­точечному принципу. У каждого маршрутизатора по три интерфейса: по одному для каждого двухточечного соединения и один — для широкове­щательного канала, который напрямую соединяет маршрутизатор с па­рой хостов. Какие подсети здесь есть? Их три: 223.1.1.0/24, 223.1.2.0/24 и 223.1.3.0/24, они напоминают подсети, которые мы видели на рис. 4.15. Но обратите внимание, что в данном примере есть и три дополнительные подсети. Одна из них, 223.1.9.0/24, включает интерфейсы, соединяющие маршрутизаторы Ml и М2; другая, 223.1.8.0/24 — интерфейсы, соединя­ющие маршрутизаторы М2 и МЗ. Наконец, третья подсеть 223.1.7.0/24 включает в себя интерфейсы, соединяющие маршрутизаторы МЗ и Ml.

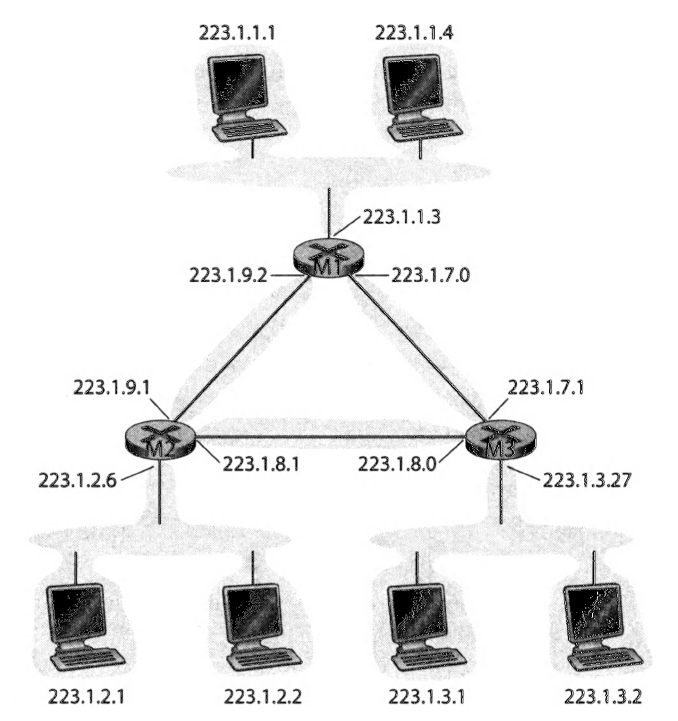
**Рис. 4.16. Адреса подсетей**

При работе с обычными взаимосвязанными системами из маршру­тизаторов и хостов мы можем определять подсети в системе по следую­щему принципу:

*Для определения границ подсетей открепите каждый интерфейс от его хоста или маршрутизатора, создав «островки» — изолированные сети, конечными точками которых являются интерфейсы. Каждая из таких изолированных сетей будет называться* ***подсетью.***

Если применить этот принцип к взаимосвязанной системе, изобра­женной на рис. 4.17, получим шесть «островков» или подсетей.

Из вышесказанного понятно, что такая организация сетей (напри­мер, в большой корпорации или вузе) со множеством Ethernet-сегментов и двухточечных соединений предполагает существование множества подсетей. У всех устройств, относящихся к какой-то из них, будет оди­наковый адрес подсети. В принципе, адреса разных подсетей могут су­щественно отличаться, но на практике они обычно весьма схожи. Чтобы понять, почему так происходит, давайте рассмотрим, как адресация об­рабатывается в Интернете на глобальном уровне.

****

**Рис. 4.17. Три маршрутизатора, объединяющие шесть подсетей**

Стратегия присваивания адресов в Интернете называется **бесклас­совая междоменная маршрутизация** (Classless Interdomain Routing, CIDR). Технология CIDR обобщает представления об адресации в подсетях. Как и при работе с подсетями 32-разрядный IP-адрес де­лится на две части и имеет точечно-десятичную форму представления *а.б.в.г/х,* где *х* указывает количество разрядов в первой части адреса.

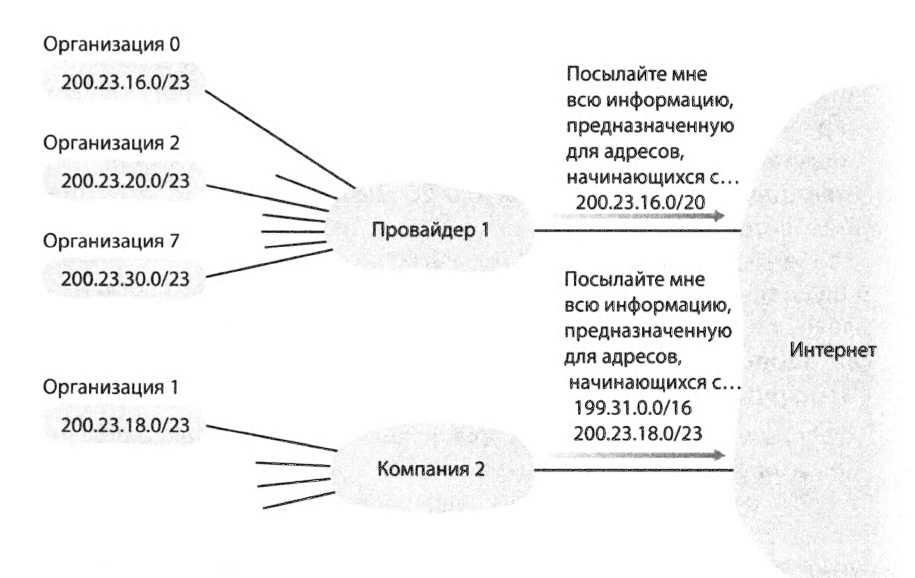
Эти *х* старших разрядов адреса, записанного в форме *а.б.в.г/х,* со­ставляют сетевую часть IP-адреса, которая часто именуется его **префик­сом** *{сетевым префиксом)* адреса. Обычно организации присваивается сплошной непрерывный блок адресов с общим префиксом. В данном случае IP-адреса устройств во всей организации будут совместно использовать общий префикс. Позднее мы обсудим действующий в Интернете протокол маршрутиза­ции BGP и убедимся, что лишь эти лидирующие *х* разрядов префикса учитываются маршрутизаторами, работающими вне сети конкретной организации. Таким образом, если действующий вне организации марш­рутизатор пересылает дейтаграмму на адрес, расположенный внутри организации, то такой внешний маршрутизатор будет учитывать лишь первые *х* разрядов адреса. Соответственно, размер таблицы маршрути­зации в таких устройствах значительно уменьшается; ведь будет доста­точно *всего одной* записи в формате *а.б.в.г/х,* чтобы направлять пакеты на *любой* адрес в пределах организации.

**ПРИНЦИПЫ В ДЕЙСТВИИ**

Ниже рассмотрен гипотетический Интернет-провайдер, подключа­ющий к Интернету восемь организаций. Этот пример наглядно ил­люстрирует, как правильно выделенные CIDR-образные адреса по­могают оптимизировать маршрутизацию. Предположим (рис. 4.18), что Интернет-провайдер, которого мы условно назовем «Провай­дер 1» объявляет всему миру, что нашей конторе следует посылать лишь такие дейтаграммы, первые 20 разрядов в адресе которых со­ответствуют шаблону 200.23.16.0/20. Таким образом, всему миру становится известно, что адресный блок 200.23.16.0/20 объединяет в себе восемь организаций, каждая из которых обладает собствен­ной подсетью. Такая возможность использовать единственный пре­фикс для объявления множества сетей часто именуется **агрегаци­ей адресов** (или **агрегацией маршрутов,** или **суммированием маршрутов).**

Агрегация адресов работает превосходно, когда IP-адреса целыми блоками выделяются провайдерам Интернет-услуг, а провайдеры, в свою очередь, раздают эти адреса клиентским организациям. Но что делать, если распределение адресов не происходит иерар­хически? Что случится, если провайдер 1 приобретет компанию 2, сделав ее своим филиалом, а затем Организация 1 подключится к Интернету именно через компанию 2? На рис. 4.18 показано, что филиал компании 2 владеет блоком адресов 199.31.0.0/16, но, к со­жалению, IP-адреса Организации 1 не относятся к этому блоку. Как решить подобную проблему? Разумеется, Организация 1 могла бы перенумеровать все свои маршрутизаторы и хосты таким образом, чтобы все новые адреса относились к адресному блоку компании 2.

Рис. 4.18. Иерархическая адресация и агрегация маршрутов

Рис. 4.19. Компания 2 предоставляет более точный маршрут к Организации 1

Однако это дорогостоящее решение, кроме того, не исключено, что в будущем обслуживание Организации 1 будет перепоручено другому филиалу провайдера 1. Как правило, реализуется вариант, при котором IP-адреса Организации 1 остаются в пространстве 200.23.18.0/23. В таком случае, как показано на рис. 4.19, про­вайдер 1 продолжает объявлять адреса из блока 200.23.16.0/20, а компания 2 — из блока 199.31.0.0/16. Однако компания 2 теперь *вдобавок* объявляет и те адреса, которые относятся к блоку Организации 1, то есть, 200.23.18.0/23. Когда другие маршрутизаторы в большом Интернете фиксируют блоки адресов 200.23.16.0/20 (Провайдер 1) и 200.23.18.0/23 (Компания 2), после чего им требу­ется переслать информацию на адрес из блока 200.23.18.0/23, такие маршрутизаторы будут искать совпадение наиболее длинного пре­фикса и направлять информацию в Компанию 2, так как эта компания объявляет самый длинный адресный префикс, максимально полно совпадающий с адресом назначения.

Соответственно, если маршрутизатор, расположенный за пределами организации, пересылает дейтаграмму, чей адрес назначения находится внутри организации, то при передаче требуется учитывать лишь *х* веду­щих бит в адресе. В результате, размер таблицы перенаправления в та­ких маршрутизаторах значительно уменьшается, поскольку всего одной записи в формате *а.б.в.г/х* будет достаточно, чтобы направить пакеты по любому адресу в рамках организации.

Остаток 32-разрядного адреса можно использовать для идентифи­кации разных устройств *внутри* организации, все они имеют одинако­вый сетевой префикс. Именно эти разряды будут учитываться при пере­сылке пакетов на маршрутизаторах *внутри* организации. Такие разряды низшего порядка могут иметь (или не иметь) дополнительную структу­ру подсетей, подобную той, что рассматривалась выше. Предположим, например, что первый 21 разряд CIDR-адреса а.б.в.г/21 указывает сете­вой префикс организации и, соответственно, является общим для всех IP-адресов у всех устройств в данной организации. Оставшиеся 11 раз­рядов далее идентифицируют конкретные хосты в рамках организации. Ее внутренняя структура может позволять использовать 11 крайних правых разрядов для построения подсетей в организации по вышеопи­санному принципу. Например, адрес а.б.в.г/24 будет относиться к кон­кретной подсети внутри организации.

До принятия стандарта CIDR длина сетевых фрагментов IP-адресов ограничивалась 8, 16 или 24 битами. Такая схема адресации называется **адресацией на основе классов,** где подсети с адресами в 8, 16 и 24 бит относились, соответственно, к классам 1, 2 и 3. Требование, в соответ­ствии с которым подсетевая часть IP-адреса обязательно должна быть равна 1, 2 или 3 байт, оказалось довольно трудновыполнимым в условиях, когда потребовалось поддерживать стремительно растущее количе­ство организаций с небольшими и средними подсетями. Подсеть класса С (/24) могла вместить в себя лишь 28 — 2 - 254 хоста (28 - 256, но два адреса резервируются для специального использования) — для многих организаций такое количество оказывается слишком малым. С дру­гой стороны, сеть класса В (/16), включающая в себя до 65 634 адре­сов, будет слишком велика. При использовании адресации на основе классов организация с 2000 хостов обычно получала адрес подсети класса В (/16). В результате это привело к быстрому израсходованию адресных пространств класса В и нерациональному использованию уже присвоенных адресных пространств. Например, если организация име­ла всего 2000 хостов, но располагала подсетью класса В, допускающей наличие вплоть до 65 534 интерфейсов, то более 63 000 адресов остава­лись неиспользуемыми, но при этом не могли быть выделены другим организациям.

Наш рассказ был бы неполон без упоминания еще одного типа IP-адреса. Речь идет о широковещательном адресе 255.255.255.255. Если хост отправляет дейтаграмму на адрес 255.255.255.255, то сообщение приходит на все хосты, расположенные в данной подсети. Предусмо­трен вариант, при котором маршрутизаторы пересылают такое сообще­ние и во все соседние подсети (хотя так обычно не делается).

Итак, мы подробно изучили IP-адресацию. Теперь давайте исследу­ем, откуда берутся адреса хостов и подсетей. Для начала рассмотрим, как организация получает блок адресов для своих устройств, затем по­говорим о том, как конкретному устройству в организации (например, хосту) присваивается адрес из этого блока.

**Получение блока адресов**

Чтобы получить блок адресов для использования в подсети кон­кретной организации, администратор вычислительной сети может для начала связаться со своим провайдером Интернета. Этот провайдер вы­делит администратору набор адресов из более крупного блока, которым располагает сам. Например, конкретному провайдеру Интернета может быть выделен блок адресов 200.23.16.0/20. Провайдер, в свою очередь, может разделить этот большой блок на восемь непрерывных блоков меньшего размера и распределить их между восемью организациями, обслуживанием которых занимается. Для удобства читателя мы под­черкнули в каждом адресе ту часть, которая относится к подсети.

**Блок адресов**

**провайдера 200.23.16.0/20 11001000 0001011100010000 00000000**

**Организация 0 200.23.16.0/23 11001000 00010111 00010000 00000000**

**Организация 1 200.23.18.0/23 11001000 00010111 00010010 00000000**

**Организация 2 200.23.20.0/23 11001000 00010111 00010100 00000000**

**••• ••• •••**

Организация 7 200.23.30.0/23 11001000 0001011100011110 00000000

Итак, можно получить адресное пространство в виде цельного бло­ка, выделенного провайдером Интернет-услуг, но это не единственный способ. Разумеется, должна быть предусмотрена возможность получе­ния адресов и для самого провайдера. Существует ли какой-нибудь гло­бальный орган, который несет полную ответственность за управление IP-адресами и за выдачу блоков адресов провайдерам Интернет-услуг и другим организациям? Конечно есть! Управление IP-адресами нахо­дится в компетенции ICANN — Интернет-корпорации по присвоению имен и номеров. Документ ICANN 2012 основан на рекомендациях, изложенных в стандарте RFC 2050. Роль некоммерческой организации ICANN, согласно документу NTIA 1998 Национального управления информации и телекоммуникаций США, заключается не только в рас­пределении IP-адресов, но и в управлении корневыми серверами систе­мы DNS. Кроме того, организация ICANN занимается очень деликатной работой по присвоению доменных имен и урегулированию споров, ко­торые могут при этом возникать. Служба распределяет адреса между ре­гиональным Интернет-регистраторами, среди которых — организации ARIN (Северная Америка), RIPE (Европа, Ближний Восток, Централь­ная Азия), APNIC (Азия и Тихоокеанский регион), LACNIC (Латинская Америка и Карибский регион). Вместе все эти учреждения образуют Организацию поддержки адресов (ASО) в составе ICANN28 и отвечают за распределение Интернет-адресов в своих регионах и за управление этими адресами.

**Получение адреса хоста: протокол динамической конфигурации хостов**

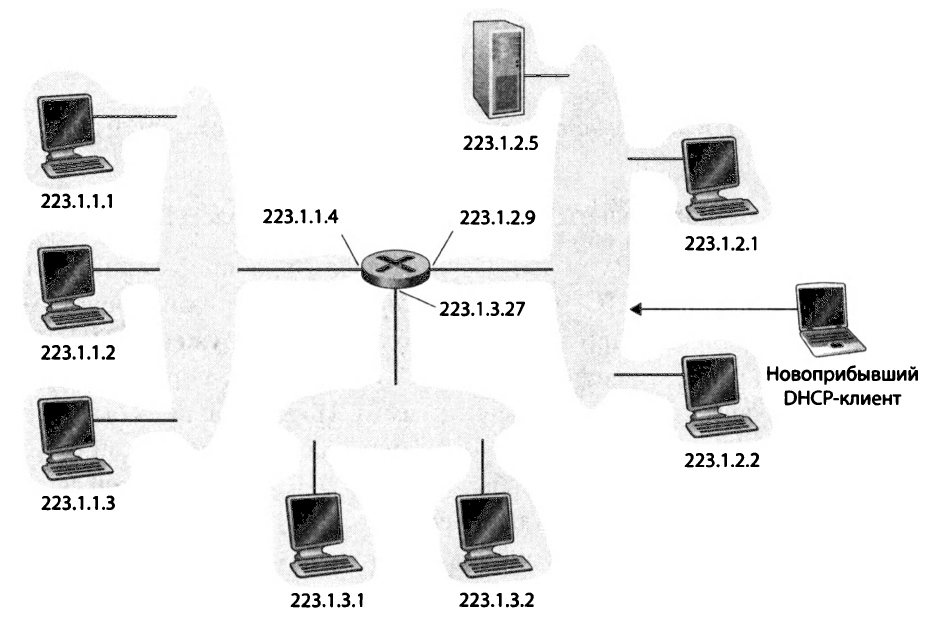
Когда организация получит блок адресов, она может далее при­сваивать отдельные IP-адреса хостам и интерфейсам маршрутизаторов в своей сети. Как правило, администратор вычислительной сети само­стоятельно конфигурирует IP-адреса в маршрутизаторе (зачастую это делается удаленно, при помощи специального сетевого инструмента).

Адреса хостов также можно сконфигурировать вручную, но обычно эта задача решается при помощи **протокола динамического конфи­гурирования хостов** (Dynamic Host Configuration Protocol, DHCP). Протокол DHCP позволяет хосту автоматически получать IP-адрес. Администратор вычислительной сети может настроить DHCP так, что конкретный хост будет получать один и тот же IP-адрес всякий раз при подключении к сети. Альтернативный вариант предусматривает при­сваивание хосту **временного IP-адреса,** который будет меняться при каждом подключении хоста к сети. Наряду с присваиванием IP-адреса хосту, протокол DHCP позволяет получать дополнительную информа­цию, в частности: маску подсети, адрес ее первого маршрутизатора на пути доставки (first-hop router), также нередко называемого «шлюз по умолчанию» (default gateway) и адрес локального DNS-сервера.

Поскольку протокол DHCP позволяет автоматизировать связанные с собственно сетью аспекты включения хоста в сеть, этот протокол за­частую называют **plug-and-play (самонастраивающимся).** Такая воз­можность делает его *исключительно* привлекательным для администра­тора вычислительной сети, которому в противном случае пришлось бы выполнять все задачи такого рода вручную! Кроме того, DHCP очень широко используется как в стационарных сетях доступа к Интернету, так и в беспроводных локальных сетях, где хосты часто подключают­ся к сети и не менее часто из нее выходят. Предположим, есть студент, который выходит в Интернет с ноутбука. Он постоянно носит его с со­бой и подключается к сети то из общежития, то из библиотеки, то из аудитории. Вполне вероятно, что в каждом из этих мест студент будет подключаться к новой подсети; соответственно, и в общежитии, и в биб­лиотеке, и в аудитории ему понадобится новый IP-адрес. Протокол DHCP идеально подходит для такой ситуации, поскольку в упомяну­тых помещениях постоянно сменяются пользователи, а выделяемый IP-адрес нужен каждому из них в течение ограниченного времени. Схо­жим образом протокол DHCP применяется различными провайдерами Интернета и в сетях стационарного доступа к Интернету. Рассмотрим, например, такого провайдера, который обслуживает 2000 абонентов, но одновременно в сеть выходят не более 400 из них. В таком случае нет необходимости в блоке из 2048 адресов; DHCP-сервер, который дина­мически присваивает пользователям адреса, может работать с блоком всего из 512 адресов (например, этот блок может иметь вид а.б.в.г/23). По мере подключения хостов к сети и отключения их DHCP-серверу требуется обновлять свой список доступных IP-адресов. Всякий раз, когда в сеть входит новый хост, DHCP-сервер выделяет ему произвольный адрес, взятый из пула тех адресов, которые свободны в настоящий момент. При выходе хоста из сети тот адрес, который ему принадлежал, возвращается в пул.

DHCP — это клиент-серверный протокол. Клиент — это, как пра­вило, новоприбывший хост, которому требуется получить информацию о конфигурации сети, в частности, IP-адрес для этого хоста. В простей­шем случае каждая подсеть (в контексте адресации, см. рис. 4.17) бу­дет иметь свой DHCP-сервер. Если в подсети нет отдельного DHCP-сервера, то его заменяет агент-ретранслятор DHCP (как правило, это маршрутизатор), которому известен (на случай необходимости) адрес DHCP-сервера данной сети. На рис. 4.20 показан DHCP-сервер, при­крепленный к подсети 223.1.2/24. Здесь маршрутизатор выступает в качестве агента-ретранслятора для подключающихся клиентов, от­носящихся к подсетям 223.1.1/24 и 223.1.3/24. В следующих разделах предполагается, что в подсети имеется свой DHCP-сервер.

Для новоприбывшего хоста DHCP-протокол выполняется в четыре этапа, как показано на рис. 4.21. Соответствующая конфигурация сети представлена на рис. 4.20. На данном рисунке значение «Ваш адрес» указывает адрес, выделяемый новоприбывшему клиенту.

**Рис. 4.20.** Клиент-серверный сценарий с применением DHCP

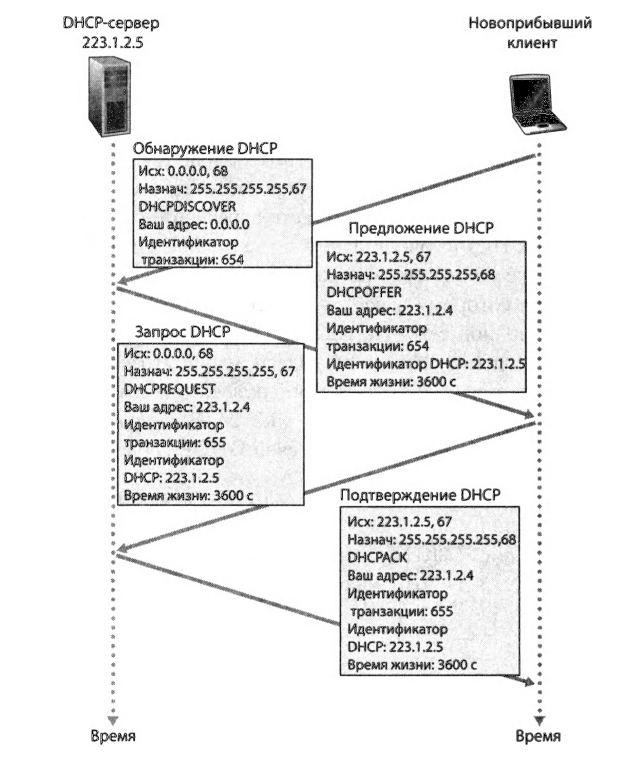


Рис. 4.21. Взаимодействие клиента и сервера по протоколу DHCP

Ниже представлены четыре этапа работы с протоколом DHCP.

*Обнаружение DHCP-cepвepa.* Первым делом новоприбывший хост должен найти тот DHCP-сервер, с которым он будет взаимодейство­вать. Это делается при помощи **сообщения обнаружения DHCP.** Это сообщение клиент отсылает в UDP-пакете на порт 67. UDP-пакет инкапсулируется в IP-дейтаграмму. Но куда должна быть от­правлена эта дейтаграмма? Хост не знает даже IP-адреса той сети, к которой он подключен, тем более адреса DHCP-сервера этой сети. В такой ситуации DHCP-клиент создает IP-дейтаграмму с сообще­нием обнаружения DHCP, широковещательным IP-адресом получа­теля 255.255.255.255 и IP-адресом отправителя «этот хост» (0.0.0.0). DHCP-клиент передает IP-дейтаграмму на канальный уровень, который затем широковещательно рассылает данный кадр на все узлы, входящие в подсеть.

* *Предложение (я) DHCP-сервера.* DHCP-сервер, получающий со­общение обнаружения, отсылает клиенту в ответ на него **сообще­ние с предложением DHCP,** которое широковещательно передает­ся всем узлам подсети. Опять же, для этого используется IP-адрес 255.255.255.255. (предлагаю подумать: а почему этот серверный от­клик также нужно передавать широковещательным образом?) По­скольку в подсети могут находиться сразу несколько DHCP-серве-ров, клиент может оказаться в завидном положении, позволяющем выбрать наиболее подходящий сервер. Каждое сообщение с пред­ложением сервера содержит транзакционный идентификатор полу­ченного сообщения обнаружения, предлагаемый клиенту IP-адрес, маску подсети и **время аренды адреса** — период, в течение которого данный IP-адрес будет валиден. Как правило, сервер предоставляет время аренды от нескольких часов до нескольких дней.
* *DHCP-запрос.* Новоприбывший клиент получает серверное предло­жение или выбирает одно из имеющихся. В качестве отклика на вы­бранное предложение клиент отсылает **сообщение запроса DHCP,** передавая вместе с ним параметры конфигурации.
* *DHCP-подтверждение (АСК).* Сервер реагирует на сообщение за­проса DHCP своим **сообщением подтверждения запроса (АСК),**

подтверждая запрошенные параметры.

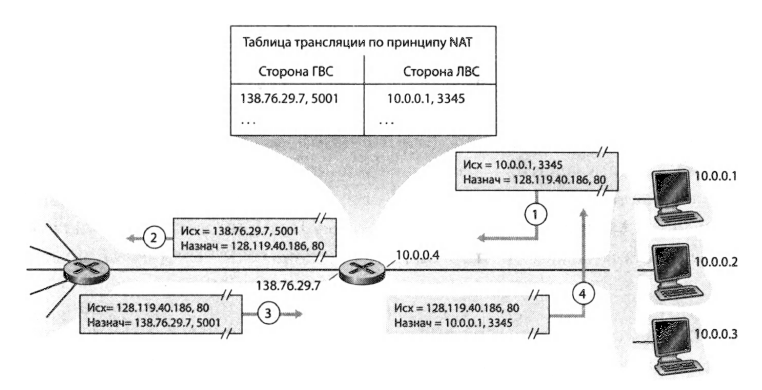
После того как клиент получит DHCP АСК, взаимодействие с сер­вером завершено, и клиент может использовать IP-адрес, выделенный по протоколу DHCP, на протяжении установленного времени аренды. Поскольку пользователю может потребоваться этот адрес и после исте­чения срока аренды, в протоколе DHCP также предусмотрен механизм, позволяющий обновить аренду.

Ценность подхода «подключи и работай» при работе с DHCP оче­видна, учитывая тот факт, что альтернативный вариант требует вручную сконфигурировать IP-адрес хоста. Вернемся к примеру со студентом, который переходит с ноутбуком из общежития в библиотеку, из библио­теки в аудиторию и всякий раз подключается к новой подсети — соот­ветственно, получает новый IP-адрес. Просто невозможно себе предста­вить, как администратор вычислительной сети в принципе мог бы сам переконфигурировать ноутбуки в каждой из таких новых точек. Кроме того, лишь немногие студенты (кроме тех, кто изучал компьютерные сети) обладают достаточным опытом, чтобы вручную сконфигурировать собственный ноутбук. В мобильных сетях, однако, DHCP имеет ряд не­достатков. Поскольку DHCP выдает новый IP-адрес всякий раз, когда узел подключается к новой подсети, мы не можем поддерживать ТСР-соединение с удаленным приложением, ведь узел переходит из одной подсети в другую. Позднее мы поговорим о мобильных IP — сравни­тельно новом усовершенствовании IP-инфраструктуры, позволяющем мобильному узлу использовать единый перманентный адрес даже при перемещении между подсетями. Дополнительные подробности о про­токоле DHCP изложены в книге Дромса и публикации Dynamic Host Configuration. Свободно распространяемая справочная реализация протокола DHCP предоставляется Консорциумом Интернет-систем.

**Трансляция сетевых адресов**

Итак, мы поговорили об адресах Интернета и о формате дейтаграмм IPv4. Очевидно, что для каждого устройства, поддерживающего связь по IP, требуется свой IP-адрес. При повсеместном распространении под­сетей для малых офисов и домашних офисов (SOHO) такая необходи­мость, казалось бы, подразумевает: всякий раз, когда в офисе формата SOHO нужно настроить локальную сеть для соединения множества устройств, нам не обойтись без диапазона адресов, который мы должны получить от Интернет-провайдера для всех наших машин. Если подсеть начнет расти (например, у наших детей появятся не только собственные ПК, но и смартфоны, и консоли Game Boy с выходом в сеть), то нам придется расширить имеющийся блок адресов. Что же делать, если наш провайдер уже раздал все непрерывные участки того адресного про­странства, в котором работает сеть нашего офиса SOHO? Более того, разве средний домовладелец умеет (или готов научиться) управлять IP-адресами своей домашней сети? К счастью, существует более про­стой способ выделения адресов, который все шире распространяется именно в таких ситуациях, как описанная здесь. Этот подход называется **трансляцией сетевых адресов** (network address translation, NAT).

На рис. 4.22 продемонстрирована работа маршрутизатора с поддерж­кой NAT. Такой маршрутизатор, установленный дома, имеет интерфейс, входящий в состав домашней сети (на рис. 4.22 она изображена справа). Адресация в домашней сети организуется точно так, как и в предыдущих примерах — все четыре интерфейса домашней сети имеют одинаковый адрес подсети: 10.0.0/24. Адресное пространство 10.0.0.0/8 представляет собой одну из трех частей пространства IP-адресов, зарезервированно­го для частных сетей согласно стандарту RFC 1918. В данном случае мы имеем **зону** с частными адресами, как в домашней сети на рис. 4.22. *Область частных адресов —* это сеть, чьи адреса являются значимыми лишь для устройств, расположенных в ее пределах. Чтобы понять важ­ность такой организации, обратим внимание на следующий факт: в мире существуют тысячи домашних сетей, причем многие из них используют одно и то же адресное пространство, 10.0.0.0/24.

****

**Рис. 4.22. Трансляция сетевых адресов**

Устройства в рамках конкретной домашней сети могут обменивать­ся пакетами при помощи адресации 10.0.0.0/24. Однако те пакеты, кото­рые направляются *за пределы* локальной сети, в глобальный Интернет, разумеется, не могут использовать эти адреса (ни как исходные, ни как целевые), поскольку с ними работают сотни и даже тысячи сетей. Таким образом, адреса из пространства 10.0.0.0/24 могут иметь значение лишь в рамках конкретной домашней сети. Но если частные адреса не имеют значения за пределами домашней сети, как же обрабатывается адреса­ция при получении пакетов из глобального Интернета или при отправке их туда — ведь в Интернете все адреса должны быть уникальны? Чтобы дать ответ на этот вопрос, требуется понимать механизм NAT.

Маршрутизатор с поддержкой NAT не *похож* на маршрутизатор, свя­зывающий сеть с внешним миром. Для внешнего мира маршрутизатор с поддержкой NAT выглядит как *единое* устройство с всего *одним* IP-адресом. На рис. 4.22 весь трафик, исходящий с домашнего маршрутизатора в глобальный Интернет имеет IP-адрес отправителя 138.76.29.7, а весь трафик, входящий в сеть через этот маршрутизатор, должен иметь адрес получателя 138.76.29.7. В сущности, маршрутизатор с поддерж­кой NAT скрывает от внешнего мира детали домашней сети. (Кстати, возможно, вы уже заинтересовались, где компьютеры из домашней сети берут себе адреса, и где домашний маршрутизатор берет свой единый IP-адрес. Зачастую ответ одинаков — DHCP! Маршрутизатор получает свой IP-адрес с DHCP-сервера провайдера Интернета, а сам маршрути­затор содержит DHCP-сервер для предоставления адресов компьюте­рам в адресном пространстве домашней сети, управляемой при помощи NAT-DHCP-маршрутизатора).

Если все дейтаграммы, прибывающие на NAT-маршрутизатор из глобальной сети, имеют один и тот же IP-адрес получателя (а имен­но — адрес интерфейса, расположенного на стыке глобальной сети и NAT-маршрутизатора), то как же маршрутизатор узнает внутренний хост, на который он должен переслать конкретную дейтаграмму? Для этого на NAT-маршрутизаторе применяется **таблица трансляции сете­вых адресов.** В записи этой таблицы включаются IP-адреса и номера портов.

Вернемся к примеру на рис. 4.22. Допустим, пользователь работает в домашней сети на хосте 10.0.0.1 и запрашивает веб-страницу с какого-либо веб-сервера (порт 80) с IP-адресом 128.119.40.186. Хост 10.0.0.1 присваивает (произвольно) номер исходного порта 3345 и посылает дейтаграмму в локальную сеть. NAT-маршрутизатор получает дейта­грамму, генерирует для нее новый номер исходного порта, на этот раз 5001, заменяет исходный IP-адрес соответствующим IP-адресом, рас­положенным на стороне ГВС (138.76.29.7) и заменяет старый номер ис­ходного порта 3345 новым — 5001. При генерировании нового номера исходного порта NAT-маршрутизатор может выбрать любой, которого пока нет в таблице трансляции сетевых адресов. (Обратите внимание: поскольку поля номеров портов 16-разрядные, протокол NAT способен поддерживать более 60 000 одновременных соединений, обходясь при этом единственным IP-адресом на стыке маршрутизатора и ГВС!) Ме­ханизм NAT в маршрутизаторе также добавляет запись в свою таблицу трансляции сетевых адресов. Веб-сервер, совершенно не представляю­щий, что прибывшая к нему дейтаграмма с HTTP-запросом уже подвер­глась обработке на NAT-маршрутизаторе, посылает в ответ дейтаграм­му, где адрес получателя — это IP-адрес NAT-маршрутизатора, а порт назначения имеет номер 5001. Когда дейтаграмма прибывает на NAT-маршрутизатор, тот делает выборку из таблицы трансляции сетевых адресов. При этом он использует целевой IP-адрес и номер порта назна­чения, чтобы получить подходящий IP-адрес (10.0.0.1) и номер порта назначения (3345) для браузера, работающего в домашней сети. Затем маршрутизатор переписывает адрес назначения дейтаграммы и номер порта назначения и пересылает ее в домашнюю сеть.

В последние годы технология NAT получила широкое распростра­нение. Однако необходимо отметить, что в сообществе IETF есть не­мало консерваторов, решительно критикующих NAT. Во-первых, гово­рят они, номера портов должны использоваться для идентификации процессов, а не хостов. (Действительно, такое отклонение от правил может вызывать проблемы в работе серверов, установленных в одной и той же домашней сети — ведь, серверные процессы ожидают входящие запросы на портах с хорошо известными номерами, которые применяются именно для этой цели.) Во-вторых, указывают критики, маршрутизаторы предназначены для обработки пакетов лишь до уровня 3 включительно. В-третьих, протокол NAT на­рушает так называемый «тезис о сквозном взаимодействии», согласно которому хосты должны обмениваться информацией непосредственно друг с другом, без посредников, изменяющих IP-адреса и номера пор­тов. Наконец, по мнению тех же критиков, мы должны работать с си­стемой IPv6 для решения проблемы с дефицитом IP-адресов, а не удо­влетворяться временными вариантами-затычками вроде NAT. Но как бы мы к этому ни относились, технология NAT становится важной со­ставляющей Интернета.

Еще одна серьезная проблема, связанная с NAT, заключается в следу­ющем: эта технология мешает работе одноранговых приложений, в част­ности, приложений для обмена файлами и для голосовой связи по IP. Как вы помните, в одноранговом приложении любой участник (пир) А должен иметь возможность инициировать ТСР-соединение с любым пиром Б. Суть проблемы заключается в том, что если Б окажется за границей NAT, то он не сможет выступить в роли сервера и принимать TCP-соединения. Проблему с NAT можно обойти, если пир А не находится за NAT. В таком случае пир А может сначала связаться с пиром Б через посредника, пира В. Пир В должен работать за пределами NAT, а также иметь установленное ТСР-соединение с пиром Б. В таком случае пир А может через пира В запросить пира Б, чтобы тот напрямую установил соединение с пиром А. Как только между пирами А и Б будет установлено прямое одноранговое TCP-соединение, два этих участника смогут обмениваться сообщениями или файлами. Такой прием называется **реверсом соединения** и действительно используется многими одноранговыми приложениями для **обхода NAT.** Если оба пира, А и Б, находятся каждый за своей границей NAT, ситуация несколько осложняется, но все равно решается при помощи ретрансляции приложений.

**Протокол UPnP**

Обход NAT все чаще осуществляется при помощи протокола UPnP для универсальной автоматической настройки сетевых устройств. Этот протокол позволяет хосту обнаружить и сконфигурировать близлежа­щий NAT-маршрутизатор. UPnP требует, чтобы и хост, и NAT были совместимы с протоколом UPnP. При его использовании UPnP при­ложение, работающее на хосте, может затребовать NAT-отображения следующей информации: *(частный IP-адрес, частный номер порта)* и *(общедоступный IP-адрес, общедоступный номер порта)* для тогопорта, номер которого мы запросили. Если NAT принимает запрос и создает отображение, то внешние узлы смогут инициировать ТСР-соединение с *(общедоступный IP-адрес, общедоступный номер порта).*Более того, протокол UPnP позволяет приложению узнать значение *(общедоступный IP-адрес, общедоступный номер порта),* так, чтоприложение сможет объявить эту информацию всему окружающему миру.

В качестве примера предположим, что наш хост, расположенный за границей NAT с поддержкой протокола UPnP, имеет частный адрес 10.0.0.1. На порте 3345 этого хоста работает BitTorrent. Кроме того, пред­положим, что общедоступный адрес этой системы NAT — 138.76.29.7. Естественно, нашему приложению BitTorrent требуется возможность приема соединений с других хостов, чтобы оно могло обмениваться с ними торрент-фрагментами. Для этого приложение BitTorrent у вас на хосте приказывает NAT создать «окно», которое будет отображать (10.0.0.1,3345) на (138.76.29.7,5001). Номер общедоступного порта 5001 выбирается приложением. Приложение BitTorrent у вас на хосте также может объявить своему трекеру, что оно доступно по адресу 138.76.29.7, 5001. Внешний хост может послать пакет TCP SYN на адрес 138.76.29.7, 5001. Когда механизм NAT получит пакет SYN, он изменит целевой IP-адрес и номер порта для этого пакета на 10.0.0.1, 3345, после чего будет выполнена переадресация пакета через NAT. Итак, протокол UPnP позволяет внешним хостам инициировать се­ансы связи с хостами, работающими через механизм NAT, при этом ис­пользуются протоколы TCP или UDP Границы NAT давно считаются настоящим бичом одноранговых приложений. Протокол UPnP, предо­ставляющий надежное и эффективное решение для обхода NAT, порой может оказаться спасительным. На этом мы вынуждены закончить наше краткое обсуждение технологий NAT и UPnP. Более подробно механизм NAT рассмотрен в работе Хьюстона и публикации How NAT Works.