В ответ на посланный узлом *C* широковещательный ARP-запрос откликается маршрутизатор с установленным протоколом Proxy-ARP. Он посылает «ложный» ARP-ответ, в котором на место аппаратного адреса компьютера *D* помещает собственный адрес MAC_1. Узел *C*, не подозревая «подвоха», посылает кадр с IP-пакетом по адресу MAC_1. Получив кадр, маршрутизатор с установленным протоколом Proxy-ARP «понимает», что он направлен не ему (в пакете указан чужой IP-адрес) и поэтому надо искать адресата в ARP-таблице. Из таблицы видно, что кадр надо направить узлу, подключенному ко второму интерфейсу.

Мы рассмотрели простейшую схему применения протокола Proxy-ARP, которая, тем не менее, достаточно полно отражает логику его работы.

Доменная служба имен DNS

Доменная служба имен (Domain Name Service, **DNS**) отображает символьные имена узлов сети на их IP-адреса (как IPv4, так и IPv6). Доменная служба имен является важной частью Интернета, но она может работать и в любой автономной IP-сети.

Пространство DNS-имен

В операционных системах, первоначально предназначенных для локальных сетей (Novell NetWare, Microsoft Windows или IBM OS/2), пользователи всегда работали с символьными именами компьютеров. Так как локальные сети состояли из небольшого числа компьютеров, применялись так называемые **плоские имена**, состоящие из последовательности символов, не разделенных на части. Примерами таких имен являются NW1_1, mail2, MOSCOW_SALES_2. Для установления соответствия между символьными именами и MAC-адресами в этих операционных системах применялся механизм широковещательных запросов, подобный механизму запросов протокола ARP. Так, широковещательный способ разрешения имен реализован в протоколе NetBIOS, на котором были построены многие локальные ОС. Так называемые NetBIOS-имена стали на долгие годы одним из основных типов плоских имен в локальных сетях.

Для стека TCP/IP, рассчитанного в общем случае на работу в больших территориально распределенных сетях, подобный подход оказывается неэффективным. В стеке TCP/IP применяется доменная система имен, которая имеет иерархическую древовидную структуру, допускающую наличие в имени произвольного количества составных частей (рис. 13.8).

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

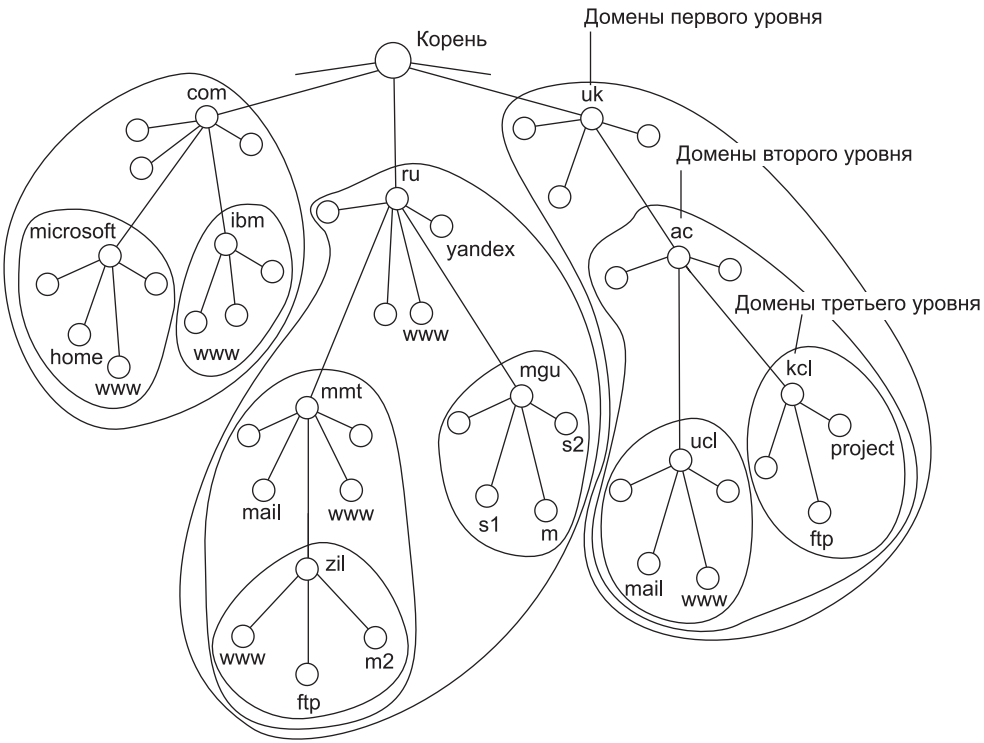

Рис. 13.8. Пространство доменных имен

Разделение имени на части позволяет *разделить административную ответственность* за назначение уникальных имен между различными людьми или организациями в пределах своего уровня иерархии. Так, для примера, приведенного на рис. 13.8, некоторая организация может нести ответственность за то, чтобы все имена с окончанием «ru» имели уникальную следующую вниз по иерархии часть. То есть все имена типа www.ru, mail.mmt.ru или m2.zil.mmt.ru отличаются второй по старшинству частью. Разделение административной ответственности позволяет решить проблему образования уникальных имен без взаимных консультаций между организациями, отвечающими за имена одного уровня иерархии. Очевидно, что должна существовать одна организация, отвечающая за назначение имен верхнего уровня иерархии.

Совокупность имен, у которых несколько старших составных частей совпадают, образуют **домен имен** (name domain). Например, имена www.zil.mmt.ru, ftp.zil.mmt.ru, yandex.ru и s1.mgu.ru входят в домен ru, так как все они имеют одну общую старшую часть — имя ru. Другим примером является домен mgu.ru. Из представленных на рис. 13.8 имен в него входят имена s1.mgu.ru, s2.mgu.ru и m.mgu.ru. Этот домен образуют имена, у которых две старшие части равны mgu.ru. Администратор домена mgu.ru несет ответственность за уникальность имен следующего уровня, входящих в домен, то есть имен s1, s2 и m. Образованные домены s1.mgu.ru, s2.mgu.ru и m.mgu.ru являются **поддоменами** домена mgu. u, так как имеют общую старшую часть имени. Часто поддомены для краткости называют только младшей частью имени, то есть в нашем случае поддоменами являются s1, s2 и m.

Как и в файловой системе, в доменной системе имен различают краткие, относительные и полные доменные имена. **Краткое доменное имя** — это имя конечного узла сети: хоста или порта маршрутизатора. Краткое имя — это лист дерева имен. **Относительное доменное имя** — это составное имя, начинающееся с некоторого уровня иерархии, но не самого верхнего. Например, www.zil — это относительное имя. **Полное доменное имя** включает составляющие всех уровней иерархии, начиная от краткого имени и кончая корневой точкой: www.zil.mmt.ru. Если в каждом домене и поддомене обеспечивается уникальность имен следующего уровня иерархии, то и вся система имен будет состоять из уникальных имен.

ВНИМАНИЕ

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

Корневой домен имен управляется центральными органами Интернета, в частности, такой организацией, как ICANN.

Домены верхнего уровня назначаются для каждой страны, а также для различных типов организаций. Для обозначения стран используются трехбуквенные и двухбуквенные аббревиатуры, например, ru (Россия), uk (Великобритания), fi (Финляндия), us (Соединенные Штаты), а для различных типов организаций, например, следующие обозначения:

- com — коммерческие организации (например, microsoft.com);
- edu — образовательные организации (например, mit.edu);
- gov — правительственные организации (например, nsf.gov);
- org — некоммерческие организации (например, fidonet.org);
- net — сетевые организации (например, nsf.net).

Каждый домен администрирует отдельная организация, которая обычно разбивает свой домен на поддомены и передает функции администрирования этих поддоменов другим организациям.

Сервер, клиент и протокол DNS

Широковещательный способ установления соответствия между символьными именами и локальными адресами, подобный реализованному в протоколе ARP, хорошо работает только в небольшой локальной сети, не разделенной на подсети. В крупных сетях, где возможность всеобщей широковещательной рассылки не поддерживается, нужен другой способ разрешения символьных имен. Альтернативой широковещательной рассылке является применение *централизованной службы*, поддерживающей соответствие между символьными именами и IP-адресами всех компьютеров сети. На раннем этапе развития Интернета на каждом хосте вручную создавался текстовый **файл отображений** с известным именем hosts.txt. Этот файл состоял из некоторого количества строк, каждая из которых содержала одну пару «доменное имя — IP-адрес», например:

```
rhino.acme.com — 102.54.94.97
```

По мере роста Интернета файлы hosts.txt также увеличивались в объеме, и создание *масштабируемого* решения для отображения имен стало необходимостью. Таким решением стала централизованная справочная служба — **система доменных имен** (Domain Name

System —**DNS**), основанная на распределенной базе отображений «доменное имя — IP-адрес».

Как и всякая сетевая служба, система DNS состоит из серверов и клиентов. **DNS-серверы** поддерживают распределенную базу отображений, а **DNS-клиенты** обращаются к серверам с запросами об отображении доменного имени на IP-адрес (эту процедуру называют также «*разрешением*» доменного имени).

DNS-клиентом является практически каждый узел Интернета, будь то клиентский компьютер, сервер приложений или маршрутизатор. Программа, реализующая функции клиента DNS, называется **резольвером**, обычно она входит в состав ОС. Приложение (например, веб-браузер), когда у него возникает необходимость в отображении доменного имени (например, имени сайта), не делает запрос напрямую к службе DNS, а обращается к резольверу своей ОС, который затем взаимодействует с DNS-сервером. Резольверу должен быть известен IP-адрес, по крайней мере одного DNS-сервера, этот адрес конфигурируется вручную или же получается по протоколу DHCP, о котором мы будем говорить в следующем разделе.

Подавляющее большинство DNS-серверов использует программное обеспечение **BIND** (Berkeley Internet Name Domain), разработанное в Калифорнийском университете Беркли.

Сервер и клиент службы DNS взаимодействуют по **протоколу DNS.** Сообщения этого протокола чаще всего передаются в *дейтаграммах UDP* с портом сервера 53, но в некоторых случаях, требующих повышенной надежности, служба DNS обращается к услугам TCP.

Иерархическая организация службы DNS

Иерархию образуют DNS-серверы. На самой вершине иерархии располагаются **корневые серверы**. Корневые серверы хранят текстовые файлы имен и IP адресов DNS- серверов следующего уровня, называемого *верхним* (top level DNS, tlDNS или TLD). Серверы верхнего уровня хранят данные об именах и адресах имен, входящих в домены верхнего уровня (com, ru или fm), а также об именах серверов DNS, которые обслуживают домены второго уровня иерархии (cisco.com, yandex.ru и т. п.).

Сервер DNS отвечает на запросы клиентов на основе информации, содержащейся в текстовых файлах отображений имен, хранящихся на данном сервере. В принципе, сервер DNS мог бы хранить данные всех отображений, входящих в некоторый домен со всеми его поддоменами; при таком подходе сервер верхнего уровня, отвечающий, например, за домен com, хранил бы в своих файлах записи всех имен, кончающихся на com: ibm.com, www.ibm.com, www2.ibm.com, cisco.com, www.cisco.com и т. д. Понятно, что такой подход не масштабируем и не может работать в глобальной сети.

Поэтому пространство доменных имен «разрезают» между серверами DNS, обычно так, чтобы сервер DNS хранил записи только в пределах одного уровня, а для имен своих поддоменов хранил только ссылки на серверы DNS, которые отвечают за эти поддомены. Например, DNS-сервер верхнего уровня, отвечающий за домен com, хранит только записи листьев своего домена, например имени www.com, а также имен серверов DNS, которые обслуживают поддомены домена com, например DNS-сервера поддомена cisco.com.

Часть пространства доменных имен, для которых некоторый сервер DNS имеет информацию об их отображениях на основе соответствующего текстового файла, называется **зоной DNS** данного сервера, а сам текстовый файл — **файлом зоны**.

Когда сервер DNS дает ответ о записи, входящей в зону, за которую он отвечает, такой ответ называется **полномочным ответом** DNS (authoritative, что можно также перевести как официальный, авторитетный или аутентичный). Как мы увидим далее, сервер DNS может также давать **неполномочный ответ**, если запрос относится не к его зоне, но он знает ответ на него за счет кэширования ответов других серверов.

Файл зоны состоит из текстовых записей нескольких типов:

- *запись типа A* — отображает имя в IPv4-адрес;
- *запись типа AAAA* — отображает имя в IPv6-адрес;
- *запись типа NS* — определяет имя DNS-сервера для некоторого домена;
- *запись типа MS* — определяет имя почтового сервера для некоторого домена,

а также некоторых других.

Протокол DNS позволяет клиенту делать запросы относительно некоторого доменного имени, задавая тип записи или же запрашивая все типы, относящиеся к данному имени. В системе DNS поддерживает не только отображение типа «один-к-одному», но и отображения «многие-к-одному» и «один-ко-многим». То сеть хост может иметь, кроме одного основного доменного имени, несколько так называемых *псевдонимов* (alias). Такое свойство оказывается полезным, например, когда владелец коммерческого сайта не хочет терять клиентов из-за того, что они могут ошибиться при наборе имени, поэтому он поддерживает в файле зоны DNS некоторый набор похожих имен, которые все отображаются на один IP-адрес.

Отображение «один-ко-многим» используется для *распараллеливания нагрузки* на веб-серверы, которые испытывают перегрузку от наплыва запросов. Для этого перегруженный сервер заменяют несколькими функционально подобными серверами. В файле зоны DNS создается запись, в которой доменному имени данного веб-сервера ставится в соответствие набор IP-адресов дублирующих серверов. В ответе на каждый запрос DNS-сервер посылает весь набор IP-адресов, предварительно сдвинув его на одну позицию, поэтому нагрузка распределяется между всеми дублирующими серверами. Для обеспечения надежности и высокой производительности для каждой зоны существует один **первичный** (primary) **сервер DNS** и несколько **вторичных** (secondary) **серверов DNS**. На первичном сервере находится исходный файл зоны — *мастер-копия файла зоны,* которая редактируется администратором сервера; вторичные серверы периодически копируют файл зоны с первичного сервера, для этого может использоваться протокол DNS, в котором имеется соответствующий тип запроса, или же администратор может использовать любой протокол копирования файлов, например, ftp или scp. Если файл зоны передается по протоколу DNS, то для повышения надежности используется *протокол TCP* (порт 53).

Итеративная и рекурсивная процедуры разрешения имени

Существуют две основные схемы разрешения DNS-имен. В первом варианте, называемом **итеративной процедурой**, работу по поиску IP-адреса координирует *DNS-клиент*: он итеративно выполняет последовательность запросов к разным серверам имен. Рассмотрим эту процедуру на примере (рис. 13.9, *а*):

1. DNS-клиент обращается к корневому DNS-серверу с указанием полного доменного имени www.zil.mmt.ru хоста, для которого он хочет найти IP-адрес.

2. Корневой DNS-сервер отвечает клиенту, указывая адреса DNS-серверов верхнего уровня, обслуживающих домен, заданный в старшей части запрошенного имени, в данном случае — домен ru.
3. DNS-клиент делает следующий запрос к одному из предложенных ему DNS-серверов верхнего уровня, который отсылает его к DNS-серверу нужного поддомена (в примере это сервер, отвечающий за зону mmt.ru), и так далее, пока не будет найден DNS-сервер, в котором хранится отображение запрошенного имени на IP-адрес. Этот сервер дает окончательный ответ клиенту, который теперь может установить связь с хостом по IP-адресу 194.85.13.5.

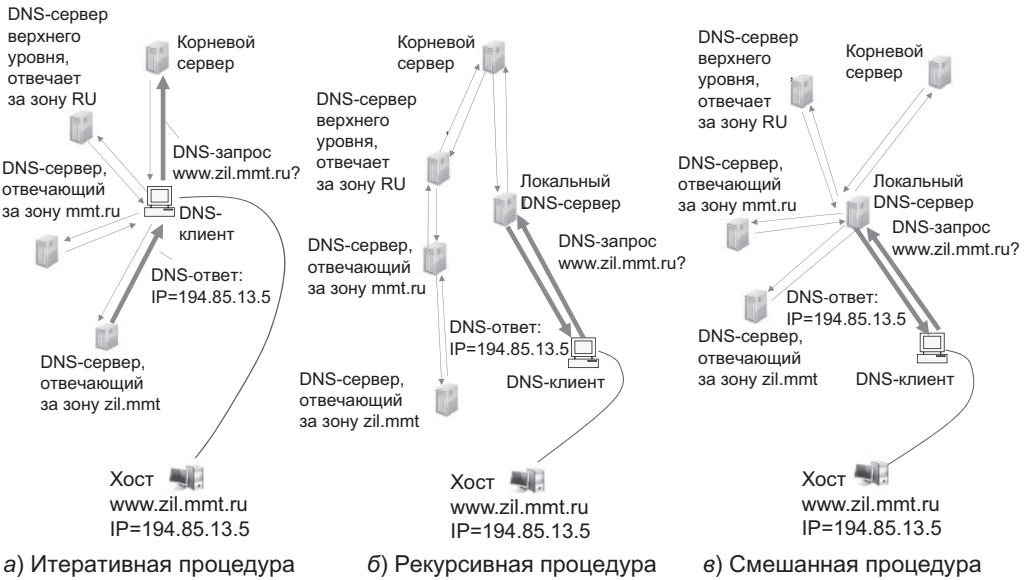

Рис. 13.9. Процедуры разрешения имен

Во втором варианте (рис. 13.9, *б*) выполняется **рекурсивная процедура**. Здесь DNS-клиент перепоручает всю работу по разрешению имени *цепочке DNS-серверов*.

1. DNS-клиент отправляет запрос к **локальному DNS-серверу**, то есть серверу, обслуживающему поддомен, которому принадлежит имя клиента.
2. Если локальный DNS-сервер знает ответ, то он сразу же возвращает его клиенту. Это может быть *полномочный* ответ (запрошенное имя входит в тот же поддомен, что и имя клиента) или *неполномочный* ответ (сервер уже узнавал данное соответствие для другого клиента и сохранил его в своем кэше).
3. Если локальный DNS-сервер не знает ответа, то он обращается к корневому серверу, который переправляет запрос к DNS-серверу верхнего уровня (отвечающему за зону RU), который в свою очередь запрашивает нижележащий сервер (зона mmt), и так далее, пока запрос не дойдет до полномочного сервера, имеющего в своем файле зоны запись о запрошенном имени.

4. DNS-ответ полномочного сервера проходит тот же путь по цепочке DNS-серверов в обратном направлении, пока не достигнет DNS-клиента, породившего данный запрос.

В третьем варианте (рис. 13.9, *в*) реализуется смешанная процедура, включающая рекурсивную и итеративную фазы:

1. Начальная часть процедуры, когда DNS-клиент передает запрос локальному DNS-серверу и поручает ему действовать от его имени, является *рекурсивной*.
2. Затем, если локальный DNS-сервер не знает ответ, то он последовательно выполняет *итеративные* запросы к иерархии серверов точно так же, как это делал DNS-клиент в первом варианте. Получив ответ, локальный DNS-сервер передает его клиенту.

DNS-серверы стараются не поддерживать рекурсивный режим ответов, так как это перегружает их; корневые серверы и серверы верхнего уровня всегда дают нерекурсивные ответы, отсылая серверы нижних уровней к серверам промежуточных уровней. Получая окончательный ответ от сервера вышестоящего уровня, рекурсивный сервер кэширует его для того, чтобы при поступлении аналогичного запроса дать быстрый неполномочный ответ. Чтобы служба DNS могла оперативно отрабатывать изменения, происходящие в сети, ответы кэшируются на относительно короткое время — обычно от нескольких часов до нескольких дней (срок задается администратором полномочного сервера). DNS-сервер может быть *открытым* (в этом случае он отвечает любому клиенту) или *закрытым* (в этом случае он отвечает либо только клиентам своего предприятия (в случае корпоративного сервера), либо только своим подписчикам услуг доступа в Интернет (в случае провайдера)).

Корневые серверы

Корневые серверы — наиболее уязвимое звено[1] службы DNS, так как разрешение всех запросов, ответы на которые не находятся в кэше или файле зоны какого-либо DNS-сервера нижнего уровня, начинаются с обращения к одному из корневых серверов. Разработчики системы DNS (в начале 80-х годов) понимали это, поэтому изначально было решено обеспечить высокую степень резервирования: было установлено 13 корневых серверов с именами a.root-servers.net, b.root-servers.net, c.root-servers.net,... m.root-servers.net и тринадцатью IP-адресами.

С тех пор организация корневых DNS-серверов изменилась. Вместо 13 серверов Интернет обслуживает более 300 — в августе 2013 года их было 376 (см. их географическое распределение на рис. 13.10). Это значительно повысило отказоустойчивость и производительность службы DNS. Все корневые серверы по-прежнему разделяют те же 13 имен (от a.root-servers.org до m.root-servers.org) и 13 IP-адресов. Но теперь каждому имени и адресу соответствует *кластер* серверов. Например, имени f.root-servers.net соответствует 56 серверов, а имени l.root-servers.net — 146. Корневые серверы распределены географически, а каждый кластер, соответствующий одному имени, администрируется отдельной организацией.

Обратная зона

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

---

[1] Об атаках и методах защиты DNS-серверов читайте в главе 29.

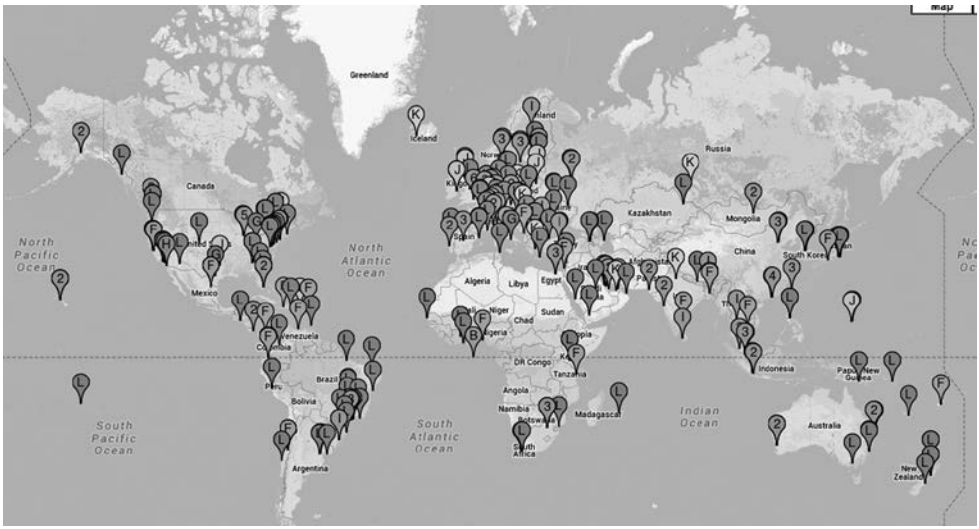

Рис. 13.10. Географическое распределение корневых серверов DNS (источник: root-servers.org)

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

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

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

Далее, учитывая, что при записи IP-адреса старшая часть является самой *левой* частью адреса, а при записи DNS-имени — самой *правой*, то составляющие в преобразованном адресе указываются в обратном порядке, то есть для данного примера — 106.31.192. Для хранения отображений всех адресов, начинающихся, например, с числа 192, заводится зона 192 со своими серверами имен. Для записей о серверах, поддерживающих старшие в иерархии обратные зоны, создана специальная зона in-addr.arpa, поэтому полная запись для использованного в примере адреса выглядит так:

```
106.31.192.in-addr.arpa
```

Серверы для обратных зон используют файлы баз данных, не зависящие от файлов основных зон, в которых имеются записи о прямом соответствии тех же имен и адресов.

Такая организация данных может приводить к несогласованности, так как одно и то же соответствие вводится в файлы дважды.

Протокол DHCP

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

Процедура присвоения адресов происходит в ходе **конфигурирования** компьютеров и маршрутизаторов. Назначение IP-адресов может происходить вручную в результате выполнения процедуры конфигурирования интерфейса, для компьютера сводящейся, например, к заполнению системы экранных форм. При этом администратор должен помнить, какие адреса из имеющегося множества он уже использовал для других интерфейсов, а какие еще свободны. При конфигурировании помимо IP-адресов сетевых интерфейсов (и соответствующих масок) устройству сообщается ряд других **конфигурационных параметров**. При конфигурировании администратор должен 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 ОС компьютер, являющийся DHCP-клиентом, посылает в сеть широковещательный запрос на получение IP-адреса. DHCP-сервер откликается и посылает сообщение-ответ, содержащее IP-адрес и ряд других конфигурационных параметров. 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[1]).

---

[1] Далее для краткости это уточнение будет опускаться.