**Протокол IMАР**

После того как Боб загрузил свои сообщения на локальный ком­пьютер, с помощью доступа по протоколу РОР3 он может создавать почтовые папки и перемещать в них загруженные сообщения. Затем он может удалять их, разносить сообщения по папкам, осуществлять поиск (по имени получателя или по теме), но размещение писем в папке на локальной машине влечет проблемы для мобильного пользователя, который предпочитает работать с иерархической структурой папок на удаленном сервере и иметь доступ к нему с любого компьютера. Такое невозможно осуществить с помощью протокола РОР3.

Чтобы решить эту и некоторые другие проблемы, был разработан протокол IMАР, описанный в документе RFC 3501. Он, как и РОР3, является протоколом доступа. Его функциональность более широкая, чем у РОР3, а, значит, более сложная, поэтому и реализация клиентских и серверных частей тоже значительно сложнее, чем у РОР3.

IMАР-сервер связывает каждое сообщение с определенным катало­гом. Когда сообщение впервые прибывает на сервер, оно привязывается к каталогу INBOX получателя (входящие сообщения). После этого по­лучатель может перемещать сообщения в новые, созданные пользовате­лем каталоги, читать, удалять и так далее. Протокол IMАР обеспечивает пользователям команды, позволяющие создавать каталоги, перемещать сообщения из одного в другой, а также производить поиск в удаленных каталогах по определенным критериям. В отличие от РОР3, IMАР-сервер обрабатывает информацию о состоянии пользователя в течение IMАР-сеанса, например имена папок и привязку сообщений к каталогам.

Еще одна важная функция протокола IMАР заключается в том, что он предоставляет команды, позволяющие пользовательскому агенту получать части сообщений, например, только заголовок, либо одну из частей МIMЕ-сообщения. Такая функциональность очень полезна, на­пример, при низкоскоростном соединении (при использовании моде­ма, например) между агентом пользователя и его почтовым сервером. В этом случае, пользователь может загружать только заголовки, избегая долгой загрузки больших сообщений, содержащих аудио- или видео­данные.

**Электронная почта через веб-интерфейс**

Все больше пользователей в сегодняшние дни отправляют или по­лучают электронную почту, используя веб-браузеры. Впервые доступ к электронной почте через веб был представлен компанией Hotmail в середине 1990-х годов. Сегодня эту услугу предлагают такие компа­нии, как Google, Yahoo! и другие крупные провайдеры. В этом случае пользовательским агентом является обычный веб-браузер, и пользова­тель взаимодействует со своим удаленным почтовым ящиком через про­токол HTTP. Когда получатель, такой как Боб, хочет получить доступ к сообщению в своем почтовом ящике, то оно с его почтового сервера отправляется в браузер Боба с помощью протокола HTTP. Протоколы РОР3 и IMАР при этом не используются. То же самое происходит, когда отправитель, такой как Алиса, хочет послать сообщение: оно передается от ее браузера к ее почтовому серверу через HTTP, а не через SMTP. При этом, однако, SMTP все же используется, когда почтовый сервер Алисы отправляет или получает сообщение с других почтовых серверов.

**DNS — служба каталогов Интернета**

Чтобы идентифицировать определенного человека, пользуются раз­ными способами. Например, мы можем идентифицировать себя именем и фамилией, номерами социального страхования, а также номером во­дительского удостоверения. Одни методы идентификации могут быть более предпочтительными по сравнению с другими. Например, ком­пьютеры налогового управления США (Internal Revenue Service, IRS) используют не имена людей, а номера социального страхования фикси­рованной длины. С другой стороны, обычные люди предпочитают использовать запоминающиеся идентификаторы (представьте, как нелепо будет выглядеть фраза: «Привет, мое имя 132-67-9875. А это мой муж, 178-87-1146»).

Так же, как и людей, хосты в Интернете можно идентифицировать несколькими способами. Один из способов идентификации — по **имени хоста.** Такие имена наглядны и нравятся людям, например, cnn.com, [www.yahoo.com](http://www.yahoo.com), [gaia.cs.umass.edu](http://gaia.cs.umass.edu) или [cis.poly.edu](http://cis.poly.edu). Однако, имена хостов, хотя и предоставляют некоторую информацию, но очень небольшую (имя [www.eurecom.fr](http://www.eurecom.fr), оканчивающееся кодом страны .fr, сообщает нам, что данный хост, вероятно, находится во Франции и не более того). Кроме того, имена хостов могут содержать разной длины буквенно-цифровые символы, что способно привести к трудностям при обработке их маршрутизаторами. По этим причинам хосты идентифицируются еще и так называемыми **IP-адресами.**

Детальнее мы обсудим IP-адреса, но здесь уместно сказать о них несколько слов. IP-адрес состоит из 4 байт и имеет строго иерар­хическую структуру. Выглядит он таким образом: 121.7.106.83, где каждая точка отделяет один байт, представленный десятичной записью от 0 до 255. IP-адрес является иерархическим, и когда мы читаем его слева направо, мы получаем все более определенную информацию о том, в каком месте Интернета располагается хост (то есть внутри какой подсети иерархической структуры сетей). Точно так же, когда мы смотрим на обычный почтовый адрес, мы получаем подробную информацию о расположении, которое этот адрес описывает.

**Службы, предоставляемые DNS**

Мы только что видели, что существует два способа идентификации хостов — по имени и по IP-адресу. Люди, например, используют мнемонические имена, которые легко запомнить, в то время как для маршрутизаторов предпочтительнее структурированные, имеющие фиксированную длину адреса. Чтобы увязать эти предпочтения, нужна какая-то служба, которая бы транслировала имена хостов в IP-адреса. Эту задачу выполняет **система доменных имен Интернета** (Domain Name System, **DNS).** DNS — это, во-первых, распределенная база данных, реализованная с помощью иерархии **DNS-серверов,** а во-вторых — протокол прикладного уровня, позволяющий хостам обращаться к этой базе данных. Часто в качестве DNS-серверов используются компьютеры с операционной системой Unix и с программным обеспечением BIND (Berkeley Internet Name Domain). Протокол DNS работает поверх UDP и использует порт 53.

DNS используется другими протоколами прикладного уровня — включая HTTP, SMTP и FTP — для транслирования переданных пользователем имен хостов в IP-адреса. В качестве примера рассмотрим, что происходит, когда браузер (то есть клиент HTTP), запущенный на хосте некоторого пользователя, запрашивает URL-адрес www.someschool.edu/index.html. Для того чтобы пользовательский хост был способен отправить HTTP-запрос веб-серверу [www.someschool.edu](http://www.someschool.edu), он должен сначала получить его IP-адрес. Это происходит следующим образом:

1. На этом хосте пользователя запускается клиентская часть приложе­ния DNS.
2. Браузер извлекает имя сервера [www.someschool.edu](http://www.someschool.edu) из URL-адреса и передает его клиентской части приложения DNS.
3. DNS-клиент отправляет запрос, содержащий это имя, серверу DNS.
4. DNS-клиент получает ответ от сервера, содержащий IP-адрес, который соответствует имени запрашиваемого веб-сервера.
5. После получения IP-адреса от клиента DNS браузер может инициировать TCP-соединение по порту 80 с процессом HTTP-сервера, находящегося по указанном адресу.

Мы видим из этого примера, что DNS вызывает дополнительные задержки для использующих его приложений (иногда очень значительные). К счастью, как мы обсудим ниже, желаемый IP-адрес очень часто уже содержится в кэше ближайшего DNS-сервера, и это уменьшает как средние задержки, так и общий сетевой трафик, вызванный DNS-запросами.

Кроме трансляции имен хостов в адреса, DNS предоставляет еще не­которые важные службы:

• **Назначение псевдонимов хостам.** Хосты с довольно сложными именами могут иметь один или несколько псевдонимов. Например, хост rе1ау1.west-coast.enterprise может иметь псевдонимы, скажем, enterprise, com и www.enterprise.com. В этом случае имя relayl.west-coast.enterprise называют **каноническим именем хоста.** Вотличие от канонических имен псевдонимы обычно более лаконичны. Приложения могут запрашивать DNS-сервер, чтобы получить не только IP-адреса, но и каноническое имя того или иного сервера по его псевдониму.

**ПРИНЦИПЫ В ДЕЙСТВИИ**

**DNS: Важные сетевые функции с помощью клиент-серверной модели**

Подобно протоколам HTTP, FTP и SMTP, DNS является протоколом прикладного уровня, так как, во-первых, он работает между взаимо­действующими конечными системами, используя клиент-серверную модель, а, во-вторых, основывается на нижележащем транспортном протоколе, чтобы передавать DNS-сообщения между этими конечными системами. Однако, с другой стороны, роль DNS отличается от веб-приложений, приложений передачи файлов или электронной почты. С DNS, в отличие от всех них, пользователь не взаимодействует напрямую. DNS обеспечивает одну из ключевых функций Интернета, а именно трансляцию имен хостов в их IP-адреса для пользовательских приложений или другого программного обеспечения Интернета. Мы отмечали ранее, что в архитектуре Интернета наиболее сложна периферия сети. Система DNS, которая реализует важный процесс преобразования имени в адрес, используя клиенты и серверы, размещенные на границе сети — это еще один пример такой клиент-серверной философии.

* **Назначение псевдонимов почтовому серверу.** По вполне понятным причинам крайне желательно, чтобы адреса электронной почты были наглядными. Например, если у Боба есть учетная запись на сервере Hotmail, то его адрес может выглядеть достаточно просто: bob@hotmail.com. Однако имя хоста почтового сервера Hotmail может быть достаточно сложным, не совсем удобным для чтения (например, когда каноническим именем хоста является что-то вроде relay1.west-coast.hotmail.com). Поэтому, почтовые приложения также могут использовать псевдонимы почтовых серверов для запросов DNS. В действительности записи MX (см. ниже) позволяют сделать так, чтобы почтовый сервер и веб-сервер какой-либо компании имели идентичные имена хостов (псевдонимы); например, веб-сервер и почтовый сервер могут оба называться enterprise.com.
* **Распределение нагрузки.** Система DNS может использоваться также для того, чтобы осуществлять распределение нагрузки между дублирующими серверами, например реплицированными веб-серверами. Достаточно нагруженные сайты, такие как, например, cnn.com, реплицируются на несколько серверов, запущенных каждый на своей конечной системе и имеющих различные IP-адреса. Для дублированных (реплицированных) веб-серверов *набор* IP-адресов привязан к одному каноническому имени хоста. Этот набор содержится в базе данных сервера DNS. Когда клиент делает запрос DNS-серверу по имени, привязанному к набору адресов, сервер в ответ выдает весь набор IP-адресов, но с каждым ответом делает ротацию их порядка. Так как клиент обычно отправляет HTTP-запрос к первому из списка адресов, то такая DNS-ротация помогает распределять трафик среди реплицированных серверов. Такая же DNS-ротация может использоваться и для почтовых серверов, которые тоже иногда имеют одинаковые псевдонимы.

Протокол DNS, описанный в документах RFC 1034 и RFC 1035, обновлен в нескольких дополнительных документах RFC. На самом деле DNS — это сложная система, и мы затрагиваем здесь только ключевые аспекты ее работы.

**Как работает DNS**

Давайте представим краткий обзор того, как же работает DNS. Рас­смотрим трансляции имени хоста в IP-адрес.

Предположим, что некоторое приложение (веб-браузер или почтовый клиент), запущенное на пользовательском хосте, требует трансляции имени хоста в IP-адрес. Приложение запускает клиентскую часть DNS, указывая имя хоста, которое нужно преобразовать. (На многих Unix-компьютерах для осуществления такой трансляции приложения вызывают команду gethostbyname().) После этого DNS пользовательского хоста отправляет сообщение-запрос в сеть. Все DNS-запросы и ответы пересылаются внутри UDP-дейтаграмм на порт 53. После некоторой задержки, варьируемой от миллисекунд до секунд, DNS пользовательского хоста получает ответное сообщение. Полученное сообщение с информацией по соответствию имени и адреса затем передается приложению. То есть, с точки зрения запущенного на пользовательском хосте приложения, DNS — это некоторый черный ящик, предоставляющий простую службу трансляции. Но в действительности этот черный ящик достаточно сложен и состоит из большого набора DNS-серверов, распределенных по всему миру, а также протокола прикладного уровня, который определяет порядок взаимодействия DNS-серверов с запраши­вающими хостами.

Самой простейшей схемой построения для службы DNS мог бы быть единственный DNS-сервер, содержащий все соответствия адресов и имен. С помощью такого централизованного сервера клиенты просто бы направляли все запросы в одно место, и DNS-сервер отвечал бы им напрямую. Несмотря на то, что такая простота очень привлекательна, в сегодняшнем Интернете с его огромным (и растущим) количеством хостов это сделать фактически невозможно. При такой централизованной организации службы могут возникать следующие проблемы:

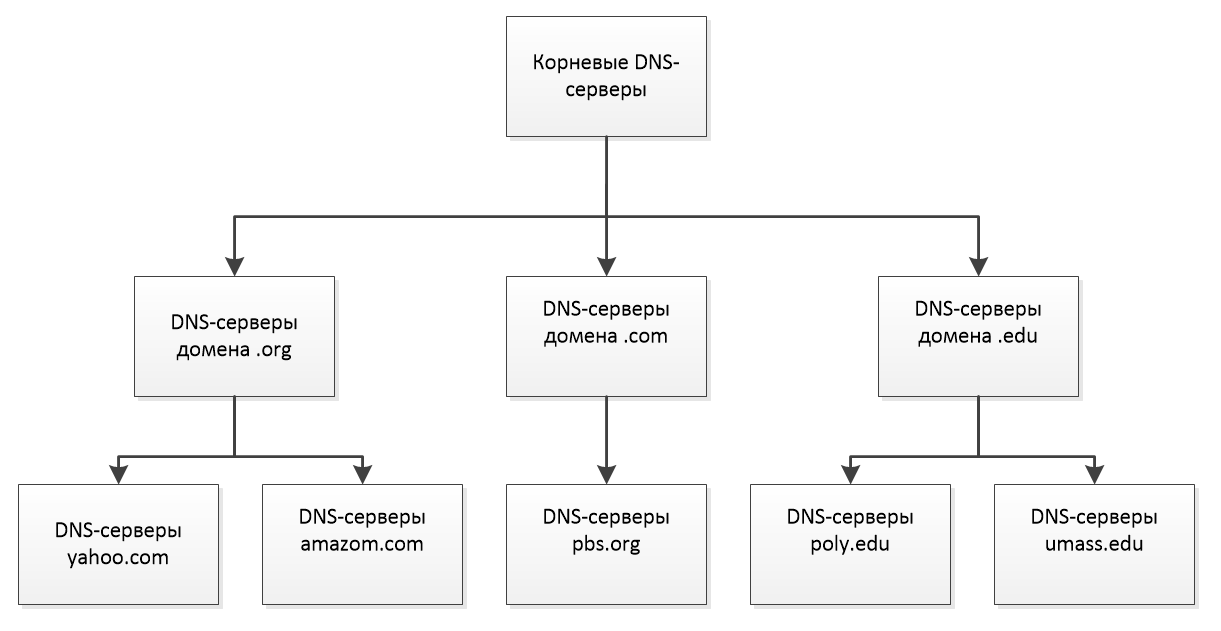
* **Единая точка отказа.** Если DNS-сервер вышел из строя, это нарушает работу всего Интернета.
* **Объем трафика.** Единственный DNS-сервер должен обрабатывать все DNS-запросы (все HTTP-запросы и сообщения электронной по­чты, сгенерированные сотнями миллионов хостов).
* **Удаленность централизованной базы данных.** Единственный DNS-сервер не может располагаться близко ко всем клиентам одновременно. Если мы расположим DNS-сервер в Нью-Йорке, то запросы, например, из Австралии будут идти с другого конца планеты и окажутся, естественно, очень медленными, что приведет к значительным задержкам.
* **Обслуживание.** Единственный DNS-сервер должен будет хранить записи для всех хостов Интернета. Это приведет к необходимости содержать огромную базу данных, которой при этом придется часто модифицироваться, чтобы учитывать каждый появляющийся новый хост.

И наконец, централизованная база данных, расположенная на един­ственном DNS-сервере, *не может быть масштабирована.* Таким обра­зом, в силу всех этих причин служба DNS строится как распределенный сервис, и она является прекрасным примером того, как распределенная база данных может быть реализована в Интернете.

**Распределенная иерархическая база данных**

Для того чтобы иметь возможность масштабирования, DNS служба использует огромный набор серверов, которые организованы в ие­рархическую структуру и распределены по всему миру. Не существует единственного DNS-сервера, содержащего таблицы соответствия для всех хостов Интернета. Вместо этого информация о соответствии адре­сов именам распределена по DNS-серверам. В первом приближении существует три класса серверов DNS — корневые DNS-серверы, DNS-серверы верхнего уровня и авторитетные (ответственные за свою зону) DNS-серверы. Их иерархическая структура показана на рис. 2.19.

Чтобы понять, как эти три класса серверов взаимодействуют, предположим, что DNS-клиент пытается определить IP-адрес для интернет-хоста [www.amazon.com](http://www.amazon.com). При этом должно происходить следующее: сначала клиент делает запрос одному из корневых серверов, который возвращает IP-адреса серверов верхнего уровня домена com. После этого клиент обращается к одному из этих серверов, который, в свою очередь, возвращает IP-адрес авторитетного сервера для amazon.com. Наконец, клиент обращается к одному из авторитетных серверов домена amazon.com, а тот возвращает IP-адрес хоста [www.amazon.com](http://www.amazon.com).

**Рис. 2.19. Часть иерархической структуры DNS-серверов**

Вскоре мы подробнее изучим процесс поисковых запросов DNS, а пока давайте посмотрим на эти три класса DNS-серверов:

* **Корневые DNS-серверы.** В Интернете существует 13 корневых DNS-серверов (обозначающиеся латинскими буквами от А до М), и большинство из них размещено в Северной Америке. Однако с внедрением технологии альтернативной адресации они «распространились» по всему миру, и фактически их число увеличилось с 13 до 123, что позволило повысить надёжность фундамента DNS. Карта узлов корневых DNS-серверов российских национальных доменов представлена на рис. 2.20; список текущих корневых серверов можно на сайте Root servers. Хотя мы ссылаемся на каждый из этих 13 серверов, как на отдельный, за каж­дым из них по соображениям безопасности и надежности стоят не­сколько реплицированных серверов.
* **DNS-серверы верхнего уровня.** Эти серверы отвечают за домены верхнего уровня, такие как com, org, net, edu, а также gov и нацио­нальные домены верхнего уровня, такие как uk, fr, ca, jp, ru и т.д. Сер­веры домена com обслуживаются компанией Verisign Global Registry Services. Серверы домена edu — компанией Educause. Полный список доменов верхнего уровня можно посмотреть в базе данных IANA
* **Авторитетные DNS-серверы.** Каждая организация, имеющая пу­блично доступные хосты (веб-серверы или почтовые серверы) в Интернете, должна предоставить также доступные DNS-записи, которые сопоставляют имена этих хостов с IP-адресами и которые находятся на авторитетном DNS-сервере. Организация может хранить эти записи либо на своем авторитетном DNS-сервере, либо на авторитетном DNS-сервере своего Интернет-провайдера. Очень многие университеты и крупные компании реализуют и обслуживают свои собственные первичные и вторичные (резервные) DNS-серверы.

**Рис. 2.20. Карта узлов корневых DNS-серверов российских национальных доменов**

Корневые, DNS-серверы верхнего уровня и авторитетные серверы — все принадлежат иерархии DNS-серверов, как показано на рис. 2.19. Но существует еще один важный тип DNS-серверов, называемый **локаль­ным DNS-сервером.** Он не обязательно принадлежит иерархии серве­ров, но, тем не менее, его роль в структуре DNS очень важна. Каждый Интернет-провайдер — верхнего или нижнего уровня — имеет локаль­ный DNS-сервер (так же называемый сервером имен по умолчанию). При подключении любого хоста к Интернету провайдер предостав­ляет владельцу хоста IP-адреса одного или нескольких своих локаль­ных DNS-серверов (обычно посредством службы DHCP, обсуждаемой позднее). Вы можете легко определить IP-адрес вашего локального DNS-сервера, просто проанализировав состояние сетевого соединения в Windows или Unix. Локальный DNS-сервер для хоста — это обычно ближайший к нему сервер. Он может даже находиться в одной локаль­ной сети с хостом. Для сетей доступа он обычно находится не более чем через несколько маршрутизаторов от пользовательского хоста. Когда хост производит DNS-запрос, этот запрос, в первую очередь, передается на локальный DNS-сервер, который действует как прокси-сервер, пере­направляя запрос в иерархию DNS-серверов. Далее мы обсудим это де­тально.

Давайте рассмотрим простой пример. Предположим, что хост [cis.poly.edu](http://cis.poly.edu) желает получить адрес хоста [gaia.cs.umass.edu](http://gaia.cs.umass.edu). Также предположим, что для первого хоста DNS-сервер называется dns.poly.edu, а авторитетный DNS-сервер для gaia.cs.umass.edu называется [dns.umass.edu](http://dns.umass.edu). Как показано на рис. 2.21, сначала хост [cis.poly.edu](http://cis.poly.edu) отправляет сообщение с DNS-запросом к своему локальному DNS-серверу, а именно [dns.poly.edu](http://dns.poly.edu). В данном запросе содержится имя хоста, которое нужно преобразовать в адрес, а именно gaia.cs.umass.edu. Локальный DNS-сервер перенаправляет сообщение-запрос на корневой DNS-сервер. Корневой DNS-сервер извлекает из записи суффикс .edu и возвращает локальному DNS-серверу список IP-адресов для серверов верхнего уровня, ответственных за домен edu. После этого локальный DNS-сервер перенаправляет запрос к одному из этих серверов верхнего уровня. Сервер верхнего уровня берет суффикс umass.edu и отвечает сообщением, содержащим IP-адрес авторитетного DNS-сервера Массачусетского университета, а именно dns.umass.edu. Наконец, локальный DNS-сервер отсылает запрос непосредственно к dns.umass.edu, который возвращает ответ с IP-адресом нужного хоста gaia.cs.umass.edu. Заметим, что в данном примере для того, чтобы получить таблицу соответствия для одного имени хоста, отправляются 8 DNS-сообщений: 4 сообщения-запроса и 4 ответных! Вскоре мы увидим, каким образом такой трафик запросов можно уменьшить с помощью DNS-кэширования.

В нашем предыдущем примере предполагалось, что DNS-сервер верхнего уровня хранит информацию об авторитетном DNS-сервере для имени целевого хоста. В общем случае, это не всегда верно. Обычно, DNS-сервер верхнего уровня может иметь информацию только о проме­жуточном DNS-сервере, которому, в свою очередь, уже известен автори­тетный DNS-сервер для имени целевого хоста. Например, предположим опять, что в Университете Массачусетса есть DNS-сервер, называемый dns.umass.edu. Еще предположим, что в каждом из отделений уни­верситета есть свой DNS-сервер, являющийся авторитетным для всех хостов своего отделения. В таком случае, когда промежуточный DNS-сервер dns.umass.edu получает запрос для хоста с именем, оканчи­вающемся на [cs.umass.edu](http://cs.umass.edu), он возвращает серверу [dns.poly.edu](http://dns.poly.edu) IP-адрес сервера [dns.cs.umass.edu](http://dns.cs.umass.edu), который является авторитетным для всех имен хостов, оканчивающихся на cs.umass.edu. Затем локальный сервер [dns.poly.edu](http://dns.poly.edu) отправляет запрос авторитетному DNS-серверу, который возвращает желаемую таблицу соответствия локальному DNS-серверу, а тот передает ее запрашивающему хосту.

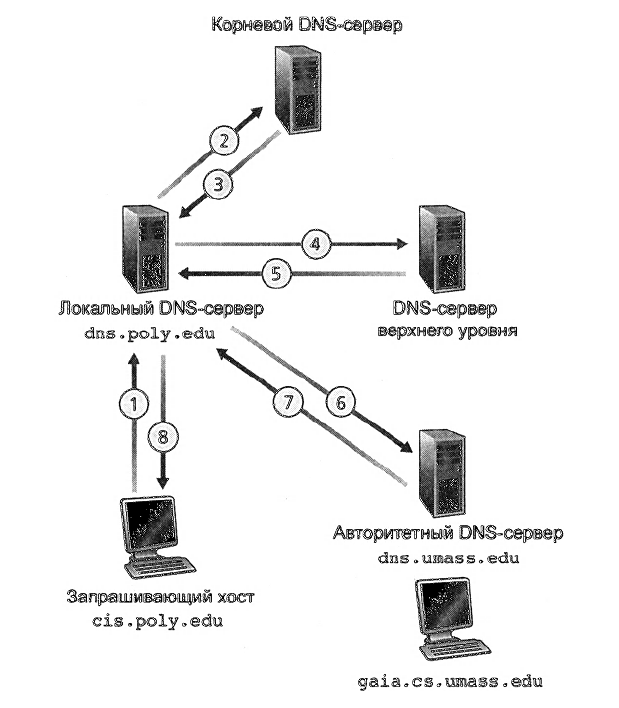
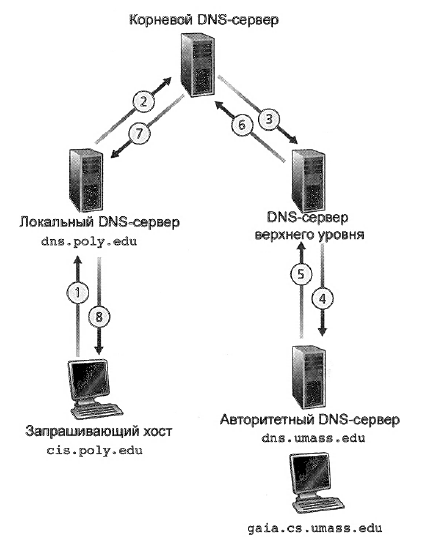


Рис. 2.21. Взаимодействие различных DNS-серверов

В этом случае будет отправлено не 8, а уже 10 DNS-сообщений! При­мер, показанный на рис. 2.21, использует как **рекурсивный,** так и **итеративный запросы.** Запрос, отправленный с cis.poly.edu к серверу [dns.poly.edu](http://dns.poly.edu), является рекурсивным, так как [dns.poly.edu](http://dns.poly.edu) получает информацию по поручению cis.poly.edu. Остальные три запроса являются итеративными, так как все три ответа непосредственно возвращаются к тому, кто запрашивал, а именно к dns.poly.edu.

Теоретически любой DNS-запрос может быть либо итеративным, либо рекурсивным. Например, рис. 2.22 показывает цепочку, в которой все запросы рекурсивные. На практике обычные запросы используют схемы, представленные на рис. 2.21: запрос от хоста к локальному DNS-серверу является рекурсивным, а все остальные запросы — итеративными.



**Рис. 2.22.** Рекурсивные запросы DNS

**DNS-кэширование**

Мы не обсудили очень важную часть системы DNS — **DNS-кэширование.** DNS-кэширование, на самом деле, очень широко используется, чтобы улучшить производительность и уменьшить число DNS-запросов, гуляющих по сети. Принцип кэширования достаточно прост. На каком-то из этапов цепочки запросов, когда DNS-сервер получает ответ (содержащий, например, таблицу соответствий имен IP-адресам), он может сохранить эту таблицу в своей локальной памяти (кэше). Например, для рис. 2.21 каждый раз, когда локальный DNS-сервер dns.poly.edu получает ответ от какого-либо другого DNS-сервера, он может кэшировать эту информацию. Если DNS-серверу поступает еще один запрос для того же самого имени хоста, DNS-сервер может предоставить желаемый IP-адрес, хранящийся у него в кэше, даже если он не является авторитетным сервером для данного имени хоста. Так как имена хостов и IP-адреса необязательно бывают постоянными, DNS-серверы обычно сбрасывают информацию в своем кэше через какой-то период времени (в некоторых случаях это составляет два дня).

В качестве примера предположим, что хост [apricot.poly.edu](http://apricot.poly.edu) запрашивает у сервера [dns.poly.edu](http://dns.poly.edu) IP-адрес хоста [cnn.com](http://cnn.com). Затем, спустя несколько часов, еще один хост Политехнического уни­верситета, скажем, [kiwi.poly.edu](http://kiwi.poly.edu) запрашивает у [dns.poly.edu](http://dns.poly.edu) то же самое имя хоста. В этом случае DNS-сервер может немедленно возвратить IP-адрес хоста cnn.com из своего кэша, не обращаясь к другим DNS-серверам. DNS-серверы способны также хранить в своем кэше IP-адреса DNS-серверов верхнего уровня, поэтому они могут очень часто выступать в качестве корневых DNS-серверов в цепочке запросов.

**Записи и сообщения DNS**

Серверы DNS образуют распределенную базу данных, где хранятся **ресурсные записи,** включая те, которые содержат таблицы соответствия имен IP-адресам. Каждый DNS-ответ несет в себе одну или более ре­сурсных записей.

Ресурсная запись представляет собой набор из четырех полей:

(Имя, Значение, Тип, Время жизни)

Время жизни определяет, когда ресурсная запись должна быть удалена из кэша. В примерах ниже мы будем игнорировать поле Время жизни. Содержание полей Имя и Значение зависит от поля Тип:

• Если Тип равен А, то в поле Имя содержится имя хоста, а в поле Значение — IP-адрес этого хоста. То есть запись типа А предоставляет стандартную таблицу преобразования имени в IP-адрес, например:

(relay1.bar.[foo.com](http://foo.com), 145.37.93.126, А). Это ресурсная запись типа А.

* Если Тип равен NS, то поле Имя является значением домена (на­пример, [foo.com](http://foo.com)), а поле Значение содержит имя хоста автори­тетного DNS-сервера, который отвечает за получение IP-адресов соответствующего домена. Такой тип записи используется, чтобы направлять DNS-запросы дальше по цепочке. Например, запись вида (foo.com, dns.foo.com, NS) является записью типа NS.
* Если Тип равен CNAME, то поле Значение содержит каноническое имя для псевдонима, который находится в поле Имя. Такая запись обеспечивает запросы канонического имени для определенного име­ни хоста. Пример такой записи: ([foo.com](http://foo.com), relay1.bar.[foo.com](http://foo.com), CNAME).

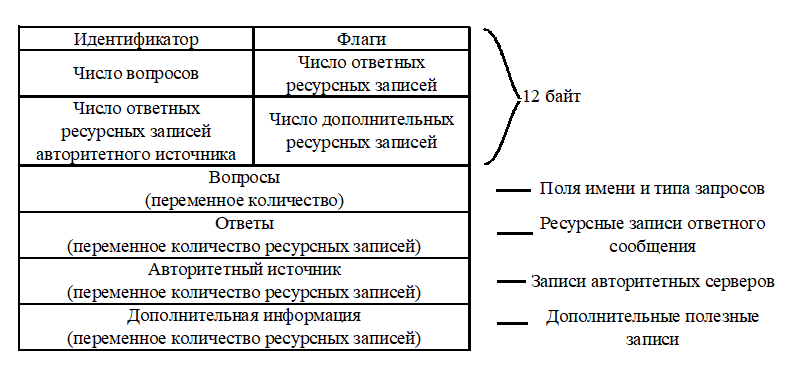
• Если Тип равен MX, то в поле Значение содержится каноническое имя почтового сервера, имеющего псевдоним, находящийся в поле Имя. Пример такой записи: ([foo.com](http://foo.com), mail.bar.[foo.com](http://foo.com), MX). Такие записи позволяют почтовым серверам иметь простые псевдонимы. Отметим, что, с помощью записи MX, компании могут использовать одинаковые псевдонимы для почтовых серверов и для веб-серверов. Чтобы получить каноническое имя почтового сервера DNS-клиент должен будет запрашивать запись MX, а если требуется каноническое имя другого сервера — запись CNAME.

Если DNS-сервер является авторитетным для какого-то определенного имени хоста, то он содержит запись типа А для этого имени хоста (он может содержать запись типа А в своем кэше, даже если не является авторитетным). Если сервер не является авторитетным для имени хоста, тогда он будет содержать запись типа NS для домена, который включает это имя хоста; а также запись типа А, в которой находится IP-адрес DNS-сервера, содержащегося в поле Значение записи типа NS. В качестве примера предположим, что DNS-сервер верхнего уровня домена edu не является авторитетным для хоста gaia.cs.umass.edu. В таком случае этот сервер будет содержать запись для домена, который включает хост [gaia.cs.umass.edu](http://gaia.cs.umass.edu), например, ([umass.edu](http://umass.edu), dns.umass.edu, NS). Он также будет иметь у себя запись типа А, которая содержит соответствие адреса DNS-сервера [dns.umass.edu](http://dns.umass.edu) IP-адресу, например, (dns .[umass.edu](http://umass.edu), 128.119.40.111, A).

**DNS сообщения**

Существуют только два типа DNS-сообщений — DNS-запрос и DNS-ответ. Мы уже упоминали эти понятия ранее. Оба типа имеют одинаковый формат, представленный на рис. 2.23. Значения различных полей DNS-сообщений следующие:

* Первые 12 байт являются *секцией заголовка* и содержат несколько полей. В первом из этих полей находится 16-разрядное число, идентифицирующее запрос. Этот идентификатор копируется в ответное сообщение, позволяя клиенту сопоставить полученный отклик с отправленным запросом. В поле «флаги» находится несколько флагов. Однобитный флаг «запрос/ответ» указывает, является ли сообщение запросом (0) или ответом (1). Однобитный флаг «авторитетный ответ» устанавливается в ответном сообщении, если DNS сервер является авторитетным для запрашиваемого хоста. Еще один однобитный флаг «требуется рекурсия» устанавливается, когда клиент (хост или другой DNS-сервер) требует от сервера обработать этот запрос самому (рекурсивный запрос). Однобитный флаг «рекурсия возможна» устанавливается в 1 в отклике DNS-сервера в случае, если он поддерживает рекурсию. В заголовок также включены 4 числовых поля, указывающих количество секций данных четырех различных типов, которые идут после заголовка.
* *Секция запросов* содержит информацию о выполняемом запросе, которая включает, во-первых, поле имени, содержащее запрашиваемое имя хоста, и, во-вторых, поле типа, которое указывает тип запроса, например, адрес, связанный с именем (тип А) или имя почтового сервера (тип MX).
* В ответе DNS-сервера *секция откликов* содержит ресурсные записи для запрашиваемых имен. Напомним, что каждая ресурсная запись содержит тип (например, A, NS, CNAME и MX), значение и время жизни. DNS-ответ способен возвращать несколько ресурсных записей, так как имя хоста может быть привязано не к одному IP-адресу, например, для реплицированных веб-серверов, обсуждаемых ранее.
* Секция *серверов имен* содержит записи других авторитетных серверов.
* Секция дополнительной информации включает другие полезные записи. Например, в отклике на запрос MX содержится ресурсная запись, представляющая каноническое имя хоста почтового сервера. В секции дополнительной информации будет содержаться запись типа А, в которой находится IP-адрес для канонического имени почтового сервера

**Рис. 2.23. Формат сообщений DNS**

Каким же образом вы можете отправить DNS-запрос непосредственно с вашего рабочего компьютера какому-либо DNS серверу? Это легко сделать, используя **программу *nslookup,*** доступную как на Windows, так и на UNIX-платформах. Например, на хосте с операционной системой Windows откройте командную строку и просто наберите *nslookup.* После этого вы можете отправить DNS-запрос любому DNS-серверу (корневому, авторитетному или серверу верхнего уровня). После получения ответа nslookup отобразит включенные в него записи (в читаемом формате). В качестве альтернативы вы можете использовать какой-либо веб-сервер, позволяющий удаленно выполнять процедуру nslookup (вы можете найти его в поисковых машинах).

**Добавление записей в базу данных DNS**

Выше мы обсудили, каким образом вызываются записи из базы данных DNS. Вам, должно быть, интересно, как они в эту базу попадают. Давайте посмотрим, как это делается, на примере. Предположим, вы создали новую компанию, которая называется Network Utopia. Первым делом вы захотите зарегистрировать доменное имя networkutopia.com в компании-регистраторе. Регистратор — это коммерческая организация, которая проверяет уникальность доменных имен, добавляет их в базу данных DNS и берет за свои услуги определенную плату. До 1999 года единственным регистратором для доменов com, net и org была компания Network Solutions. В настоящие дни множество компаний-регистраторов конкурируют между собой. Они должны получать аккредитацию **в *корпорации по управлению доменными именами и IP-адресами (Internet Corporation for Assigned Names and Numbers, ICANN).*** Полныйсписок аккредитованных регистраторов можно найти на сайте **www.**[**internic.net**](http://internic.net)**.**

Когда вы регистрируете домен networkutopia.com у какого-то регистратора, вам необходимо предоставить ему имена и IP-адреса ваших первичного и вторичного DNS-серверов. Предположим, что эти адреса и имена таковы: dns1.networkutopia.com, dns2.networkutopia.com, 212.212.212.1 и 212.212.212.2. Регистратор должен убедиться, что для этих двух авторитетных DNS-серверов записи типа NS и типа А добавлены в DNS-серверы верхнего уровня домена com. Например, для первичного авторитетного сервера [networkutopia.com](http://networkutopia.com) регистратор добавит следующие две ресурсные записи в систему DNS:

(networkutopia.com, dns1.networkutopia.com, NS)

(dns1.networkutopia.com, 212.212.212.1, A)

Вы также должны убедиться, что в ваши авторитетные DNS-серверы добавлены ресурсные записи для вашего веб-сервера www. networkutopia.com (запись типа А) и для вашего почтового сервера mail.networkutopia.com (запись типа MX). До недавнего времени содержимое каждого из DNS-серверов настраивалось статически с помощью конфигурационного файла, создаваемого администратором вычислительной сети. Не так давно появилась опция в протоколе DNS для обновления базы данных через DNS-сообщения.

**О БЕЗОПАСНОСТИ**

**Уязвимости DNS**

Мы узнали, что служба DNS является одним из важнейших компо­нентов инфраструктуры Интернета. Многие важные службы — в том числе веб-приложения и электронная почта — просто не в состоянии функционировать без DNS. Поэтому возникает естественный во­прос, может ли DNS подвергнуться нападению? Является ли служба DNS беспомощной мишенью, ожидающей, когда выведут из строя ее и вместе с ней большинство интернет-приложений?

**Первый тип атаки, который приходит на ум, это DDoS-атака на сер­вер DNS с переполнением полосы пропускания. Например, злоумышленник может попытаться отправить каждому корневому DNS-серверу огромный поток пакетов — настолько значительный, что большинство законных DNS-запросов никогда не будут обработаны. Такая масштабная DDoS-атака на корневые серверы DNS, в действительности, имела место 21 октября 2002 года, когда злоумышленники использовали ботнет для отправки огромного потока ICMP-сообщений каждому из 13 корневых серверов DNS. (На данный момент достаточно знать, что ICMP-пакеты — это специальные виды IP-дейтаграмм.) К счастью, это крупномасштабное нападение привело к минимальным последствиям и мало или вообще никак не отразилось на работе пользователей Интернета. Нападавшие смогли направить большой поток пакетов на корневые DNS-сервера, но многие из них были защищены пакетными фильтрами, настроенными на блокировку ICMP-трафика. В результате они успешно перенесли атаку и продолжили функционировать в нормальном режиме. Помогло и то, что большинство локальных серверов DNS кэшируют IP-адреса серверов верхнего уровня и заменяют собой корневые серверы при запросах.**

**Потенциально более эффективной DDoS-атакой против DNS была бы отправка потока DNS-запросов к серверам верхнего уровня, например, всем обрабатывающим домен .com, так как, во-первых, DNS-запросы сложнее фильтровать, а во-вторых, для локальных серверов DNS обрабатывать запросы за серверы верхнего уровня сложнее, чем за корневые. Но и в этом случае кэширование в ло­кальных серверах DNS значительно бы уменьшило тяжесть послед­ствий.**

**Служба DNS может быть атакована и другими способами. При атаке «человек посередине» (MITM-атака) злоумышленник перехватывает запросы от хостов и возвращает фиктивные ответы. В такой раз­новидности атаки, как «отравление DNS», злоумышленник посы­лает поддельные ответы DNS-серверу, заставляя сервер принять фиктивные записи в свой кэш. Любой из этих двух видов атак может быть использован для перенаправления ничего не подозревающего пользователя на веб-сайт злоумышленника. Эти атаки, однако, достаточно трудно осуществить, так как они требуют перехвата па­кетов или «удушения» (снижения пропускной способности) серве­ров.**

**Другой важной разновидностью является не атака на службу DNS как таковая, а использование инфраструктуры DNS для проведения DDoS-атаки против целевого хоста (например, почтового сервера вашего учебного заведения). При этом атакующий передает DNS-запросы на многие авторитетные серверы DNS, и в каждом запросе подставляет адрес целевого хоста в качестве поддельного источника запроса. DNS-серверы затем отправляют свои ответы непосредственно на целевой хост. Если запросы построить таким образом, чтобы размер ответа был гораздо больше (в байтах) размера самого запроса (так называемое усиление DNS), то злоумышленник может потенциально подавить хост-жертву с помощью трафика от DNS-северов без необходимости генерировать свой собственный. Такие атаки с использованием «DNS-отражения» имели некоторый ограниченный успех.**

**Можно сделать вывод, что система DNS продемонстрировала себя на удивление устойчивым к атакам сервисом. На сегодняшний день не было совершено ни одной атаки, которая бы парализовала работу этой службы.**

**Атаки с использованием «DNS-отражения» можно рассматривать лишь как угрозу определенной конфигурации DNS-серверов.**

После того как все эти шаги будут завершены, пользователи Интер­нета смогут посещать ваш веб-сайт и отправлять электронную почту сотрудникам вашей компании. Давайте закончим обсуждение DNS, проверив, что это действительно так. Эта проверка также поможет закрепить наши знания о DNS. Предположим, Алиса, находясь в Австралии, хочет посмотреть во Всемирной паутине страницу [**www.networkutopia.com**](http://www.networkutopia.com)**.** Как уже говорилось ранее, ее хост сначала отправит DNS-запрос ее локальному серверу DNS. Локальный сервер DNS, в свою очередь, свяжется с сервером верхнего уровня домена com. (Локальный сервер DNS обратится к корневому DNS-серверу, если в своем кэше не обнаружит адрес сервера верхнего уровня.) Этот сервер верхнего уровня содержит ресурсные записи типов NS и А, перечисленные выше, поскольку регистратор добавил эти записи во все серверы верхнего уровня домена com. Он отправляет их в ответном сообщении локальному DNS-серверу Алисы. После этого локальный сервер DNS отправляет сообщение серверу с адресом 212.212.212.1, запрашивая запись типа А, соответствующую имени [**www.networkutopia.com**](http://www.networkutopia.com)**.** Эта запись предоставляет IP-адрес нужного веб-сервера, скажем, 212.212.71.4, который локальный сервер DNS передает обратно хосту Алисы. Теперь браузер Алисы может инициировать TCP-соединение с хостом 212.212.71.4 и отправлять запрос HTTP через это соединение. Вот так! Когда кто-то путешествует по сети, кроме видимых глазу событий в его браузере, за кадром происходит еще много интересных вещей!