**Domain Name System (DNS)**

1. **Протокол DNS** относится к протоколам стека TCP/IP (прикладной уровень).

Описан в RFC 1034 (Concepts and Facilities), 1035(implementation and specification) или STD 13.

**Служба DNS (домены, зоны; зоны прямого и обратного просмотра; основные и дополнительные зоны; рекурсивный и итеративный запросы на разрешение имен).**

***Историческая справка:*** *Систему доменных имен разработал в 1983 году Пол Мокапетрис. Тогда же было проведено первое успешное тестирование DNS, ставшей позже одним из базовых компонентов сети Internet. С помощью DNS стало возможным реализовать масштабируемый распределенный механизм, устанавливающий соответствие между иерархическими именами сайтов и числовыми IP-адресами.*

*В 1983 году Пол Мокапетрис работал научным сотрудником института информатики (Information Sciences Institute, ISI), входящего в состав инженерной школы университета Южной Калифорнии (USC). Его руководитель, Джон Постел, предложил Полу придумать новый механизм, устанавливающий связи между именами компьютеров и адресами Internet, — взамен использовавшемуся тогда централизованному каталогу имен и адресов хостов, который поддерживала калифорнийская компания SRI International.*

*«Все понимали, что старая схема не сможет работать вечно, — вспоминает Мокапетрис. — Рост Internet становился лавинообразным. К сети, возникшей на основе проекта ARPANET, инициированного Пентагоном, присоединялись все новые и новые компании и исследовательские институты».*

*Предложенное Мокапетрисом решение — DNS — представляло собой распределенную базу данных, которая позволяла организациям, присоединившимся к Internet, получить свой домен.*

*«Как только организация подключалась к сети, она могла использовать сколь угодно много компьютеров и сама назначать им имена», — подчеркнул Мокапетрис. Названия доменов компаний получили суффикс .com, университетов — .edu и так далее.*

*Первоначально DNS была рассчитана на поддержку 50 млн. записей и допускала безопасное расширение до нескольких сотен миллионов записей. По оценкам Мокапетриса, сейчас насчитывается около 1 млрд. имен DNS, в том числе почти 20 млн. общедоступных имен. Остальные принадлежат системам, расположенным за межсетевыми экранами. Их имена неизвестны обычным Internet-пользователям.*

*Новая система внедрялась постепенно, в течение нескольких лет. В это время ряд исследователей экспериментировали с ее возможностями, а Мокапетрис занимался в ISI обслуживанием и поддержанием стабильной работы «корневого сервера», построенного на мэйнфреймах компании Digital Equipment. Копии таблиц хостов хранились на каждом компьютере, подключенном к Internet, еще примерно до 1986 года. Затем начался массовый переход на использование DNS.*

*Необходимость отображения имен сетевых узлов в IP-адреса*

Компьютеры и другие сетевые устройства, отправляя друг другу пакеты по сети, используют IP-адреса.. Однако пользователю (человеку) гораздо проще и удобнее запомнить некоторое символические имена сетевых узлов, чем четыре бессодержательных для него числа. Однако, если люди в своих операциях с сетевыми ресурсами будут использовать имена узлов, а не IP-адреса, тогда должен существовать механизм, сопоставляющий именам узлов их IP-адреса.

Есть два таких механизма — локальный для каждого компьютера файл *hosts* и централизованная иерархическая служба имен *DNS*.

1. **ICANN** (Internet Corporation for Assign Names and Numbers) – организация, обеспечивающая управление всем адресным пространством Internet.

*Использование локального файла hosts и системы доменных имен DNS для разрешения имен сетевых узлов*

На начальном этапе развития сетей, когда количество узлов в каждой сети было небольшое, достаточно было на каждом компьютере хранить и поддерживать актуальное состояние простого текстового файла, в котором содержался список сетевых узлов данной сети. Список устроен очень просто — в каждой строке текстового файла содержится пара «IP-адрес — имя сетевого узла». В системах семейства Windows данный файл расположен в папке “*%system root%\system32\drivers\etc*” (где *%system root%* обозначает папку, в которой установлена операционная система). Сразу после установки системы Windows создается файл hosts с одной записью «127.0.0.1 localhost».

С ростом сетей поддерживать актуальность и точность информации в файл hosts становится все труднее. Для этого надо постоянно обновляя содержимое этого файла на всех узлах сети. Кроме того, такая простая технология не позволяет организовать пространство имен в какую-либо структуру. Поэтому появилась необходимость в централизованной базе данных имен, позволяющей производить преобразование имен в IP-адреса без хранения списка соответствия на каждом компьютере. Такой базой стала DNS (Domain Name System) — система именования доменов, которая начала массовую работу в 1987 году.

Заметим, что с появлением службы DNS актуальность использования файла host совсем не исчезла, в ряде случаев использование этого файла оказывается очень эффективным.

*Служба DNS: пространство имен, домены*

DNS — это иерархическая база данных, сопоставляющая имена сетевых узлов и их сетевых служб IP-адресам узлов. Содержимое этой базы, с одной стороны, распределено по большому количеству серверов службы DNS, а с другой стороны, является централизованно управляемым. В основе иерархической структуры базы данных DNS лежит *доменное пространство имен* (*domain namespace*), основной структурной единицей которого является домен, объединяющий сетевые узлы (хосты), а также поддомены. Процесс поиска в БД службы DNS имени некого сетевого узла и сопоставления этому имени IP-адреса называется «разрешением имени узла в пространстве имен DNS».

Служба DNS состоит из трех основных компонент:

* **Пространство имен DNS и соответствующие ресурсные записи (RR, resource record)** — это сама распределенная база данных DNS;
* **Серверы имен DNS** — компьютеры, хранящие базу данных DNS и отвечающие на запросы DNS-клиентов;
* **DNS-клиенты (DNS-clients, DNS-resolvers)** —компьютеры, посылающие запросы серверам DNS для получения ресурсных записей.

Пространство имен.

Пространство имен DNS — иерархическая древовидная структура, начинающаяся с корня, не имеющего имени и обозначаемого точкой “**.**”. Схему построения пространства имен DNS лучше всего проиллюстрировать на примере сети Интернет (Рис. 3.8).

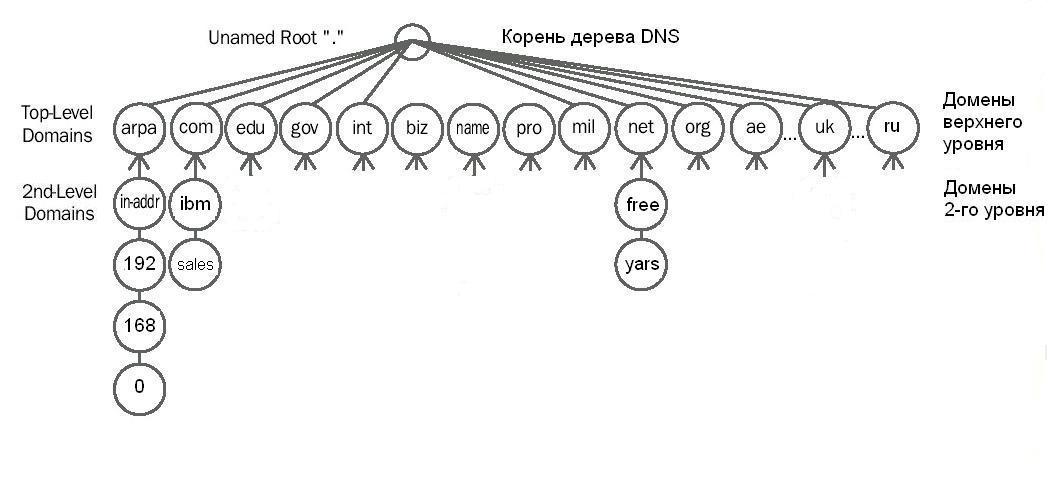


Рис 3.8

Для доменов 1-го уровня различают 3 категории имен:

* **ARPA** — специальное имя, используемое для обратного разрешения DNS (из IP-адреса в полное имя узла);
* **Общие (generic) имена 1-го уровня** — 16 (на данный момент) имен, назначение которых приведено в таблице 3.4;
* **Двухбуквенные имена для стран** — имена для доменов, зарегистрированных в соответствующих странах (например, *ru* — для России, *ua* — для Украины, *uk* — для Великобритании и т.д.).

|  |  |
| --- | --- |
| **Имя домена** | **Назначение** |
| aero | Сообщества авиаторов |
| biz | Компании (без привязки к стране) |
| com | Коммерческие организации, преимущественно в США (например, домен *microsoft.com* для корпорации Microsoft) |
| coop | Кооперативы |
| edu | Образовательные учреждения в США |
| gov | Правительственные учреждения США |
| info | Домен для организаций, предоставляющих любую информацию для потребителей |
| int | международные организации (например, домен *nato.int* для НАТО) |
| mil | Военные ведомства США |
| museum | Музеи |
| name | Глобальный домен для частных лиц |
| net | Домен для Интернет-провайдеров и других организаций, управляющих структурой сети Интернет |
| org | Некоммерческие и неправительственные организации, преимущественно в США |
| pro | Домен для профессиональных объединений (врачей, юристов, бухгалтеров и др.) |
| job | Кадровые агентства |
| travel | Туроператоры |

Корневые серверы DNS — DNS-серверы, содержащие информацию о доменах верхнего уровня, указывающую на DNS-серверы, поддерживающие работу каждого из этих доменов. Основные корневые серверы DNS размещены в домене root-servers.net обозначаются латинскими буквами от A до М. Они управляются различными организациями, действующими по согласованию с ICANN. Количество серверов ограничено в связи с максимальным объёмом UDP-пакета (большее количество серверов потребовало бы перехода на TCP-протокол для получения ответа, что существенно бы увеличило нагрузку).

Хотя из-за ограничения на размеры DNS-пакета (512 байт) в DNS-ответ может быть помещено всего 13 серверов, на самом деле сейчас за этими 13 «виртуальными» серверами стоят 190 серверов (зеркала).

По разным оценкам только от 18 до 32 % разрешений доменных имен приводит к обращению непосредственно к одному из корневых серверов, остальные запросы используют кэшированные DNS-записи о TLD NS.

Домен верхнего (первого) уровня (англ. top-level domain — TLD) — в иерархии системы доменнных имён (DNS) самый высокий уровень после корневого домена (англ. root domain). Является начальной точкой отсчёта (справа налево), с которой начинается доменное имя в Интернете.

Полное имя узла (FQDN, fully qualified domain name) состоит из нескольких имен, называемых метками (label) и разделенных точкой. Самая левая метка относится непосредственно к узлу, остальные метки — список доменов от домена первого уровня до того домена, в котором находится узел (данный список просматривается справа налево).

FQDN (сокр. от англ. Fully Qualified Domain Name, полностью определённое имя домена, иногда сокращается до «полное имя») — имя домена, не имеющее неоднозначностей в определении. Включает в себя имена всех родительских доменов иерархии DNS.

В DNS и, что особенно существенно, в файлах зоны(англ.), FQDN завершаются точкой (например, example.com.), то есть включают корневое доменное имя, которое является безымянным.

Максимальный размер FQDN — 255 байт, с ограничением в 63 байта на каждое имя домена.

Серверы имен DNS.

Серверы имен DNS (или DNS-серверы) — это компьютеры, на которых хранятся те части БД пространства имен DNS, за которые данные серверы отвечают, и функционирует программное обеспечение, которое обрабатывает запросы DNS-клиентов на разрешение имен и выдает ответы на полученные запросы.

DNS-клиенты.

DNS-клиент — это любой сетевой узел, который обратился к DNS-серверу для разрешения имени узла в IP-адрес или, обратно, IP-адреса в имя узла.

*Служба DNS: домены и зоны*

Как уже говорилось выше, каждый DNS-сервер отвечает за обслуживание определенной части пространства имен DNS. Информация о доменах, хранящаяся в БД сервера DNS, организуется в особые единицы, называемые *зонами* (*zones*). Зона — основная единица репликации данных между серверами DNS. Каждая зона содержит определенное количество ресурсных записей для соответствующего домена и, быть может, его поддоменов.

Системы семейства Windows Server поддерживают следующие типы зон:

* **Стандартная основная** (**standard primary**) — главная копия стандартной зоны; только в данном экземпляре зоны допускается производить какие-либо изменения, которые затем реплицируются на серверы, хранящие дополнительные зоны;
* **Стандартная дополнительная** (**standard secondary**) — копия основной зоны, доступная в режиме «только-чтение», предназначена для повышения отказоустойчивости и распределения нагрузки между серверами, отвечающими за определенную зону; процесс репликации изменений в записях зон называется «*передачей зоны*» (*zone transfer*)  
  (информация в стандартных зонах хранится в тестовых файлах, файлы создаются в папке *«%system root%\system32\dns*», имя файла, как правило, образуется из имени зоны с добавлением расширения файла «*.dns*»; термин «стандартная» используется только в системах семейства Windows);
* **Интегрированная в Active Directory** (**Active Directory–integrated**) — вся информация о зоне хранится в виде одной записи в базе данных Active Directory (такие типы зон могут существовать только на серверах Windows, являющихся контроллерами доменов Active Directory; в интегрированных зонах можно более жестко управлять правами доступа к записям зоны; изменения в записях зоны между разными экземплярами интегрированной зоны производятся не по технологии передачи зоны службой DNS, а механизмами репликации службы Active Directory);
* **Зона-заглушка** (**stub**; только в Windows 2003) — особый тип зоны, которая для данной части пространства имен DNS содержит самый минимальный набор ресурсных записей (начальная запись зоны SOA, список серверов имен, отвечающих за данную зону, и несколько записей типа A для ссылок на серверы имен для данной зоны).

Рассмотрим на примере соотношение между понятиями *домена* и *зоны*. Проанализируем информацию, представленную на рис. 3.9.

**microsoft.com**

**sales.microsoft.com**

**it.microsoft.com**

**edu.microsoft.com**

**Файл:**

**“sales.microsoft.com.dns”**

**Файл:**

**“it.microsoft.com.dns”**

**Файл:**

**“microsoft.com.dns”**

Рис. 3.9

В данном примере пространство имен DNS начинается с домена *microsoft.com*, который содержит 3 поддомена: *sales.microsoft.com*, *it.microsoft.com* и *edu.microsoft.com* (домены на рисунке обозначены маленькими горизонтальными овалами). Домен — понятие чисто логическое, относящееся только к распределению имен — древовидная структура доменов и поддоменов и для каждого домена свой список узлов. Понятие домена никак не связано с технологией хранения информации о домене. Зона — это способ представления информации о домене и его поддоменах в хранилище тех серверов DNS, которые отвечают за данный домен и поддомены. В данной ситуации, если для хранения выбрана технология стандартных зон, то размещение информации о доменах может быть реализовано следующим образом:

* записи, относящиеся к доменам microsoft.com и edu.microsoft.com, хранятся в одной зоне в файле *«microsoft.com.dns»* (на рисунке зона обозначена большим наклонным овалом);
* управление доменами sales.microsoft.com и it.microsoft.com *делегировано* другим серверам DNS, для этих доменов на других серверах созданы соответствующие файлы *«sales.microsoft.com*.*dns*» и *«it.microsoft.com.dns»* (данные зоны обозначены большими вертикальными овалами).

Делегирование управления — передача ответственности за часть пространства имен другим серверам DNS.

**Первичный DNS-сервер**: сервер, который всегда может прочитать свой зонный файл. сервер, имеющий право на внесение изменений в данные зоны. Для каждой зоны может быть только один первичный сервер.

1. **Вторичный DNS**-**сервер:** сервер, получающий свои данные о первичного сервера. Он поддерживает свой зонный файл, но постоянно его обновляет. Для обновления использует запись SOA. Для каждой зоны, должен существовать вторичный сервер. Большинство реализаций обязывают вторичный сервер обновлять только изменения зоны первичного сервера. Вторичные сервера поддерживаю актуальную копию зонного файла.
2. **Кэширующий сервер:**  используется редко, в больших разветвленных организациях.

*Зоны прямого и обратного просмотра*

Зоны, рассмотренные в предыдущем примере, являются *зонами прямого просмотра* (*forward lookup zones*). Данные зоны служат для разрешения имен узлов в IP-адреса. Наиболее часто используемые для этого типы записей: A, CNAME, SRV.

Для определения имени узла по его IP-адресу служат *зоны обратного просмотра* (*reverse lookup zones*), основной тип записи в «обратных» зонах — PTR. Для решения данной задачи создан специальный домен. с именем «*in-addr.arpa*». Для каждой IP-сети в таком домене создаются соответствующие поддомены, образованные из идентификатора сети, записанного в обратном порядке. Записи в такой зоне будут сопоставлять идентификатору узла полное FQDN-имя данного узла. Например, для IP-сети *192.168.0.0/24* необходимо создать зону с именем «*0.168.192.in-addr.arpa*». Для узла с IP-адресом 192.168.0.10 и именем *host.company.ru* в данной зоне должна быть создана запись «10 PTR host.company.ru».

*Алгоритмы работы итеративных и рекурсивных запросов DNS*

Все запросы, отправляемые DNS-клиентом DNS-серверу для разрешения имен, делятся на два типа:

* итеративные запросы (клиент посылает серверу DNS запрос, в котором требует дать наилучший ответ без обращений к другим DNS-серверам);
* рекурсивные запросы (клиент посылает серверу DNS запрос, в котором требует дать окончательный ответ даже если DNS-серверу придется отправить запросы другим DNS-серверам; посылаемые в этом случае другим DNS-серверам запросы будут итеративными).

В основном DNS-клиентами используются рекурсивные запросы. На рис. 5.3 проиллюстрирован процесс разрешения доменного имени с помощью рекурсивного запроса.

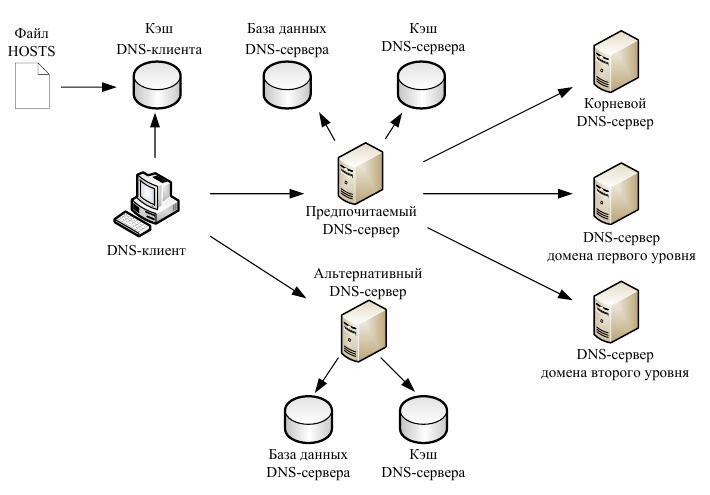


Рис. 5.3. Процесс обработки рекурсивного DNS-запроса

Сначала DNS-клиент осуществляет поиск в собственном локальном кэше DNS-имен. Это память для временного хранения ранее разрешенных запросов. В эту же память переносится содержимое файла HOSTS (каталог windows/system32/drivers/etc). Утилита IPconfig с ключом /displaydns отображает содержимое DNS-кэша. Если кэш не содержит требуемой информации, DNS-клиент обращается с рекурсивным запросом к предпочитаемому DNS-серверу (Preferred DNS server), адрес которого указывается при настройке стека TCP/IP. DNS-сервер просматривает собственную базу данных, а также кэш-память, в которой хранятся ответы на предыдущие запросы, отсутствующие в базе данных. В том случае, если запрашиваемое доменное имя не найдено, DNS-сервер осуществляет итеративные запросы к DNS-серверам верхних уровней, начиная с корневого DNS-сервера.

Рассмотрим на примерах, как происходит взаимодействие DNS-клиента и DNS-сервера при обработке итеративных и рекурсивных запросов.

Допустим, что пользователь запустил программу Обозреватель Интернета и ввел в адресной строке адрес «*http://www.microsoft.com*». Прежде чем Обозреватель установит сеанс связи с веб-сайтом по протоколу HTTP, клиентский компьютер должен определить IP-адрес веб-сервера. Для этого клиентская часть протокола TCP/IP рабочей станции пользователя (так называемый *resolver*) сначала просматривает свой локальный кэш разрешенных ранее имен в попытке найти там имя «*www.microsoft.com*». Если имя не найдено, то клиент посылает запрос DNS-серверу, указанному в конфигурации TCP/IP данного компьютера (назовем данный DNS-сервер «*локальным DNS-сервером*»), на разрешение имени «*www.microsoft.com*» в IP-адрес данного узла. Далее DNS-сервер обрабатывает запрос в зависимости от типа запроса.

Вариант 1 (итеративный запрос).

Если клиент отправил серверу итеративный запрос (напомним, что обычно клиенты посылают рекурсивные запросы), то обработка запроса происходит по следующей схеме:

* сначала локальный DNS-сервер ищет среди зон, за которые он отвечает, зону «*microsoft.com*»; если такая зона найдена, то в ней ищется запись для узла *www*; если запись найдена, то результат поиска сразу же возвращается клиенту;  
  в противном случае локальный DNS-сервер ищет запрошенное имя «*www.microsoft.com*» в своем кэше разрешенных ранее DNS-запросов; если искомое имя есть в кэше, то результат поиска возвращается клиенту;  
  если локальный DNS-сервер не нашел в своей базе данных искомую запись, то клиенту посылается IP-адрес одного из корневых серверов DNS;
* клиент получает IP-адрес корневого сервера и повторяет ему запрос на разрешение имени «*www.microsoft.com*»;  
  корневой сервер не содержит в своей БД зоны «*microsoft.com*», но ему известны DNS-серверы, отвечающие за зону «*com*», и корневой сервер посылает клиенту IP-адрес одного из серверов, отвечающих за эту зону;
* клиент получает IP-адрес сервера, отвечающего за зону «*com*», и посылает ему запрос на разрешение имени «*www.microsoft.com*»;  
  сервер, отвечающий за зону «*com*», не содержит в своей БД зоны «*microsoft.com*», но ему известны DNS-серверы, отвечающие за зону «*microsoft.com*», и данный DNS-сервер посылает клиенту IP-адрес одного из серверов, отвечающих уже за зону «*microsoft.com*»;
* клиент получает IP-адрес сервера, отвечающего за зону «*microsoft.com*», и посылает ему запрос на разрешение имени «*www.microsoft.com*»;  
  сервер, отвечающий за зону «*microsoft.com*», получает данный запрос, находит в своей базе данных IP-адрес узла *www*, расположенного в зоне *«microsoft.com»*, и посылает результат клиенту;  
  клиент получает искомый IP-адрес, сохраняет разрешенный запрос в своем локальном кэше и передает IP-адрес веб-сайта программе Обозреватель Интернета (после чего Обзреватель устанавливает связь с веб-сайтом по протоколу HTTP).

Вариант 2 (рекурсивный запрос).

Если клиент отправил серверу рекурсивный запрос, то обработка запроса происходит по такой схеме:

* сначала локальный DNS-сервер ищет среди зон, за которые он отвечает, зону «*microsoft.com*»; если такая зона найдена, то в ней ищется запись для узла *www*; если запись найдена, то результат поиска сразу же возвращается клиенту;  
  в противном случае локальный DNS-сервер ищет запрошенное имя «*www.microsoft.com*» в своем кэше разрешенных ранее DNS-запросов; если искомое имя есть в кэше, то результат поиска возвращается клиенту;
* если локальный DNS-сервер не нашел в своей базе данных искомую запись, то сам локальный DNS-сервер выполняет серию итеративных запросов на разрешение имени «*www.microsoft.com*», и клиенту посылается либо найденный IP-адрес, либо сообщение об ошибке.

*Реализация службы DNS в системах семейства Windows Server*

Главная особенность службы DNS в системах семейства Windows Server заключается в том, что служба DNS разрабатывалась для поддержки службы каталогов Active Directory. Для выполнения этой функции требуются обеспечение двух условий:

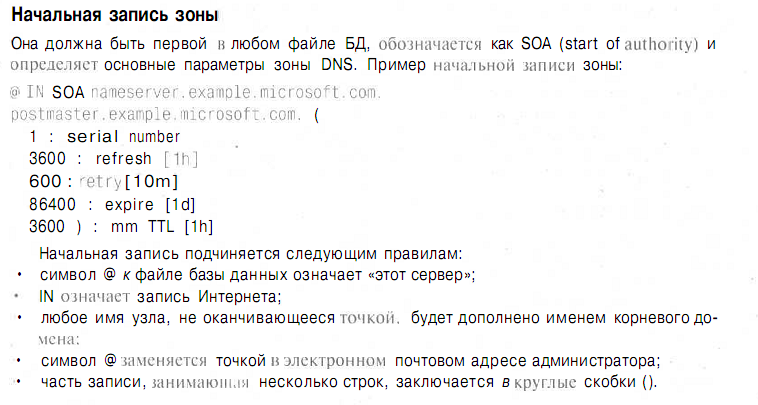
* поддержка службой DNS динамической регистрации (dynamic updates);
* поддержка службой DNS записей типа SRV.

Для непосредственного отображения пространства имен в пространство IP-адресов служат т.н. ресурсные записи (RR, resource record). Каждый сервер DNS содержит ресурсные записи для той части пространства имен, за которую он несет ответственность (authoritative). Таблица 3.5 содержит описание наиболее часто используемых типов ресурсных записей.

|  |  |  |
| --- | --- | --- |
| **Тип ресурсной записи** | **Функция записи** | **Описание использования** |
| A | Host Address  Адрес хоста, или узла | Отображает имя узла на IP-адрес  (например, для домена *microsoft.com* узлу с именем *www.microsoft.com* сопоставляется IP-адрес с помощью такой записи:  www A 207.46.199.60) |
| CNAME | Canonical Name (alias)  Какноническое имя (псевдоним) | Отображает одно имя на другое *(задаёт псевдоним который действует так же как и символьное имя)* |
| MX | Mail Exchanger  Обмен почтой | Управляет маршрутизация почтовых сообщений для протокола SMTP *(выводит почтовые адреса соотв имени)* |
| NS | Name Server  Сервер имен | Указывает на серверы DNS, ответственные за конкретный домен и его поддомены *(адрес DNS-серверов ответственных за зону)* |
| PTR | Pointer  Указатель | Используется для обратного разрешения IP-адресов в имена узлов в домене *in-addr.arpa* |
| SOA | Start of Authority  Начальная запись зоны | Используется для указания основного сервера для данной зоны и описания свойств зоны |
| SRV | Service Locator  Указатель на службу | Используется для поиска серверов, на которых функционируют определенные службы (например, контроллеры доменов Active Directory или серверы глобального каталога) *(указание на местоположение (адрес и порты) серверов для сервисов)* |

Таблица 3.5

1. **Запись SOA:** ***имя домена зоны***, ***почтовый адрес администратора***, ***порядок*** – число, увеличивается при изменении зоны; ***период обновления***  (для вторичных серверов); ***задержка перед следующей попыткой обновления*** (после неудачной попытки); ***интервал времени***, через который сервер утратит статус ответственного после обновления; ***минимальное время жизни*** показывает другим (кэширующим) серверам, как долго они могут пользоваться данными данного сервера.



1. **Запись A:**  связывает доменное имя и ip-адрес. Если создать несколько записей с одним и тем же доменным именем но разными ip-адресами, то это позволяет ***равномерно*** распределить нагрузку между несколькими хостами (***карусель***). Сервер выбирает ***ближайший*** ip-адрес к клиенту.

Вывод

nslookup -type=A localhost

A-запись

Запись типа A позволяет установить соответствие между именем хоста в домене и его IP-адресом.

Запись типа A имеет следующий формат:

имя\_хоста [TTL] A IP-адрес

Имя\_хоста: доменное имя хоста (устройства), подключенного к Интернету, для которого данная запись определяет соответствие с его IP-адресом.

А: тип записи

IP-адрес: IP-адрес хоста

Примеры A-записи для хоста info.test.ru в файле зоны test.ru:

info A 194.85.61.44

или

info.test.ru. 86400 A 194.85.61.44

1. **Запись CNAME**: позволяет одному ip-адресу сопоставить несколько доменных имен.

CNAME-запись

Запись типа CNAME (Canonical Name - Каноническое имя) позволяют присваивать хосту мнемонические имена. Мнемонические имена, или псевдонимы, широко применяются для связывания с хостом какой-либо функции, либо просто для сокращения имени.

Реальное имя иногда называют каноническим.

Если для хоста есть запись типа CNAME, которая содержит его мнемонические имена, другие записи для данного хоста должны ссылаться на его реальное (каноническое) имя, а не на мнемоническое. Когда программы DNS встречают запись CNAME, они прекращают свои запросы по мнемоническому имени и переключаются на реальное имя.

Кроме того, если данное имя использовано в качестве псевдонима, то на него нельзя занести записи любого другого типа.

Т.е. недопустима конструкция вида:

domain CNAME имя\_хоста

domain MX 10 почтовый сервер

Мнемоимена полезны, например, в случае, когда имя хоста изменилось и вы хотите разрешить пользователям, знающим старое имя, получить доступ к хосту.

Запись типа CNAME имеет следующий формат:

Мнемоимя [TTL] CNAME имя\_хоста

TTL: см. описание параметра TTL

CNAME: тип записи

имя\_хоста: каноническое имя хоста

Примеры CNAME-записей для хоста arhive.test.ru в домене test.ru.:

ftp.test.ru. CNAME arhive.test.ru.

gopher.test.ru. CNAME arhive.test.ru.

или

ftp.test.ru. 86400 CNAME arhive.test.ru.

gopher.test.ru. 172800 CNAME arhive.test.ru.

1. **Запись MX**: связывает имя почтового домена с именем хоста, содержащего почтовый сервер. Можно указать несколько MX-записей с указанием приоритета.

MX-запись

Запись типа MX (Mail Exchange - почтовый сервер) определяет почтовый сервер - машину, которая обрабатывает почту для вашего домена.

Запись типа MX имеет следующий формат:

ваш\_домен [TTL] MX приоритет почтовый сервер

TTL: см. описание параметра TTL

MX: тип записи

Приоритет: определяет значение приоритетности почтового сервера. Чем меньше число, тем выше приоритет почтового сервера (0 означает самый высокий приоритет, 65535 - самый низкий). Таким образом, почтовый сервер с более высоким приоритетом является основным, а почтовые серверы с более низкими приоритетами будут второстепенным и вступят в работу в том случае, если все более приоритетные серверы по каким-либо причинам недоступны или неработоспособны.

почтовый сервер: имя почтового сервера

Примеры MX-записей для домена test.ru:

test.ru. MX 10 relay2.test.ru.

test.ru. MX 20 relay3.test.ru.

или

test.ru. 86400 MX 10 relay2.test.ru.

test.ru. 86400 MX 20 relay3.test.ru.

Обращаем ваше внимание на то, что у всех записей типа MX, относящихся к одному домену, значение TTL должно быть одинаковым.

Таким образом, почтовый сервер relay2.test.ru является основным, а relay3.test.ru является второстепенным почтовым сервером и вступает в работу в том случае, если relay2.test.ru по каким-либо причинам недоступен или неработоспособен.

1. **Запись PTR**: запись обратная записи A (зона обратного просмотра).

PTR-запись

Записи типа PTR (Pointer - указатель) служат для выполнения обратного преобразования IP-адресов в имена хостов. Для каждого сетевого интерфейса хоста рекомендуется создать запись PTR.

Записи типа PTR, как правило, имеет смысл вносить только в обратные зоны.

Если провайдер выделил вам несколько IP-адресов из своей сети, то по поводу записей в обратной зоне вам следует обращаться к нему.

Запись типа PTR имеет следующий формат:

адрес [TTL] PTR имя\_хоста

адрес: преобразованный IP-адрес хоста

TTL: см. описание параметра TTL

PTR: тип записи

Примеры PTR-записей

Если провайдер выделил вам IP-адрес 194.85.61.42 в сети 194.85.61.0/24, то запись о вашем хосте (например, www.mydomain.ru) будет сделана провайдером в обратной зоне 61.85.194.in-addr.arpa. Запись будет иметь вид:

42 PTR www.mydomain.ru.

или

42 86400 PTR [www.mydomain.ru](http://www.mydomain.ru).

1. **Запись NS:**  определяет полномочный сервер имен субдомена. Связывает имя субдомена с именем полномочного dns-сервера.

NS-запись

Записи типа NS (Name Server - cервер имен) описывают DNS-серверы для данного домена. Количество записей типа NS в файле зоны должно точно соответствовать количеству DNS-серверов, обслуживающих домен и включать все DNS-серверы, указанные в домене. Для доменов второго уровня это DNS-серверы, указанные в полях "nserver" в информации по домену, которую можно получить, используя сервис Whois (https://www.nic.ru/whois/)

Запись типа NS имеет следующий формат:

доменное\_имя [TTL] NS имя\_хоста

TTL: см. описание параметра TTL

NS: тип записи

имя\_хоста: доменное имя DNS-сервера

Примеры NS-записей в домене test.ru. Указаны DNS-серверы, обслуживающие домен test.ru и домен третьего уровня mf в домене test.ru (mf.test.ru)

Для домена test.ru:

test.ru. NS ns3.test.ru.

test.ru. NS ns4.test.ru.

или

test.ru. 86400 NS ns3.test.ru.

test.ru. 86400 NS ns4.test.ru.

Для домена mf.test.ru:

mf.test.ru. NS ns3.test.ru.

mf.test.ru. NS ns4.test.ru.

или

mf.test.ru. 86400 NS ns3.test.ru.

mf.test.ru. 86400 NS ns4.test.ru.

Обращаем ваше внимание на то, что у всех записей типа NS, относящихся к одному доменному имени, значение TTL должно быть одинаковым.

1. **Запись SRV:** отражают порядок и доступность служб в конкретной зоне. Определяет сетевой протокол (\_kerberos, \_ldap, \_gc, \_kpasswd) , порт, доменное имя хоста.

SRV-запись

Записи типа SRV используются для поиска серверов, обеспечивающих работу тех или иных служб в данном домене.

С подробным описанием этого типа записей вы можете ознакомиться в RFC-2782.

Запись типа SRV имеет следующий формат:

\_Service.\_Proto.Name [TTL] SRV Priority Weight Port имя\_хоста

Service: название службы (пример: ldap, kerberos, gc и другие)

Proto: протокол, при помощи которого клиенты могут подключиться к данной службе (пример: tcp, udp)

Name: имя домена, в котором размещена данная служба

TTL: см. описание параметра TTL

SRV: тип записи

Priority: приоритет данного сервера. Чем меньше число, тем выше приоритет (0 означает самый высокий приоритет, 65535 - самый низкий)

Weight: относительный вес для серверов с одинаковым приоритетом. Предназначен для распределения нагрузки между серверами, для которых указан равный приоритет

Port: порт, на котором размещена указанная служба на данном сервере

Target: доменное имя сервера, предоставляющего данную службу

Примеры SRV-записей

\_foobar.\_tcp SRV 0 1 9 old-slow-box.example.com.

или

\_foobar.\_tcp 86400 SRV 0 3 9 new-fast-box.example.com.

1. **Типы запросов в DNS-серверу: *рекурсивный***, ***итеративный***.
2. **Реализации DNS-серверов**: BIND (Berkeley Internet Name Domain), Microsoft DNS Server, OpenDNS.

BIND (Berkeley Internet Name Domain, до этого: Berkeley Internet Name Daemon) — открытая и наиболее распространённая реализация DNS-сервера, обеспечивающая выполнение преобразования DNS-имени в IP-адрес и наоборот.

BIND поддерживается организацией Internet Systems Consortium. BIND был создан студентами и впервые был выпущен в BSD 4.3.

В Unix этот сервер является стандартом де-факто, но имеются и альтернативы:

PowerDNS — www.powerdns.com;

MyDNS — DNS-сервер, использующий в качестве БД MySQL;

Microsoft DNS Server — входит в состав серверных версий Windows.

***Утилита NSLOOKUP***

Утилита nslookup используется для проверки способности DNS-серверов выполнять разрешение имен. Утилита может работать в двух режимах:

• режим командной строки – обычный режим запуска утилит командной строки. Утилита nslookup выполняется в этом режиме, если указан какой-либо ключ;

• интерактивный режим – в этом режиме возможен ввод команд и ключей утилиты без повторения ввода имени утилиты.

Команды утилиты nslookup:

• help или ? – вывод справки о командах и параметрах утилиты;

• set – установка параметров работы утилиты;

• server <имя> – установка сервера по умолчанию (Default Server), используемого утилитой, с помощью текущего сервера по умолчанию;

• lserver <имя> – установка сервера по умолчанию утилиты с помощью первоначального; • root – установка сервера по умолчанию утилиты на корневой сервер;

• ls <домен> – вывод информации о соответствии доменных имен IP-адресам для заданного домена;

• exit – выход из интерактивного режима.