**DNS и доменные имена**

Как бы вы повели себя, если бы знаменитый продукт рекламировали по телевидению таким образом: "Посетите наш Web-*сервер* по адресу 198.182.196.56, чтобы получить более подробную информацию..."? Успели бы вы загрузить свой ПК и войти в *Internet* прежде, чем забудете этот *адрес*? Я бы не успел...

К сожалению, человеку неподвластно так хорошо обрабатывать числа, как это делают компьютеры. Чтобы компенсировать этот человеческий недостаток, системные администраторы стали пользоваться обычными символьными именами, которыми они стали называть компьютерные системы. Для облегчения процедуры поиска компьютера в сети *Internet* была разработана *система доменных имен* *DNS* (*Domain Name System*). *DNS* является существенным компонентом в процессе обработки электронной почты. Для себя вы можете решить, что все вопросы, касающиеся вашего домена и электронной почты, будут в ведении провайдера услуг сети *Internet*. В таком случае *знание* подробностей о работе *DNS* для вас не столь критично, хотя в принципе представлять, как работает *DNS*, не помешает (хотя бы на случай возникновения каких-либо неполадок). В этой лекции мы рассмотрим, как появилась система доменных имен, почему она является одним из основных компонентов при обработке электронной почты и каким образом сконфигурировать *сервер* на базе ОС Linux в качестве клиента либо сервера *DNS*.

**История возникновения имен хостов**

В те времена, когда *сеть* *Internet* еще была небольшой (всего несколько сотен компьютеров), найти нужную машину в ней не представляло особого труда. Каждый *компьютер* в сети *Internet* хранил базу данных хостов и IP-адресов. Имена хостов в *Internet* могли быть каким угодно и назначались администратором, например: Fred, Barney, Acct1 и т.д. Существовал также централизованный механизм учета изменений имен хостов. Примерно раз в неделю системный *администратор* копировал текущую базу данных имен хостов к себе на *компьютер*. Конечно, эта система имела свои недостатки. При подключении нового компьютера к сети требовалось убедиться, что новое название, назначаемое хосту, еще не присвоено какой-либо другой системе. Вскоре системные администраторы поняли, что этот метод назначения имен хостов становится серьезным препятствием на пути развития сети. С ростом сети *Internet* расширялась и база данных имен хостов. А с расширением базы данных увеличивалось и время ее загрузки для получения обновлений и время поиска нужных имен в ней. Кроме того, все сложнее становилось определять хостам уникальное символьное имя. Что-то должно было прийти на смену базе данных имен. И это были ....

**Доменные имена**

Метод именования хостов, который был разработан, получил название системы доменных имен (*DNS*). В *DNS* используется иерархическая распределенная *база данных*, которая пришла на смену базе данных имен хостов. Теперь можно было бы сказать, что "с тех пор ни на одном компьютере не ведется *база данных* всех хостов сети *Internet*". На самом деле эта *база данных* распределена между различными компьютерами сети *Internet*, которые называются серверами *DNS*. Теперь один *компьютер*-клиент, чтобы найти другой *компьютер* в сети *Internet*, должен обратиться с запросом к ближайшему серверу *DNS* и получить IP-*адрес* удаленного компьютера. В порядке внедрения этой системы был разработан новый протокол для обмена информацией между сервером *DNS* и клиентом. Для серверов *DNS* было разработано соответствующее *программное обеспечение*, с помощью которого они могли обслуживать новую систему баз данных.

**Структура DNS**

Структура иерархической базы данных подобна древовидной структуре с узлами, формирующими дерево. Узел верхнего уровня называется **корнем**. Корневой узел не указывается напрямую в адресах, поэтому он еще называется безымянным узлом. В корневом уровне сформировано несколько категорий, которые делят общую базу данных на части, называемые **доменами**. В каждом домене имеются серверы DNS, ответственные за обслуживание базы данных имен хостов данной области сети (части распределенной базы данных имен). На [рис. 4.1](https://intuit.ru/studies/professional_skill_improvements/1274/courses/116/lecture/3361?page=1#image.4.1) представлена диаграмма распределения доменов в системе DNS.



**Рис. 4.1.**Система доменных имен сети Internet

Первый (или как его еще называют верхний) уровень распределения разделен на домены на основе кодов стран. Дополнительно к доменам верхнего уровня отнесены домены, созданные для различных организаций в США. Это было сделано для того, чтобы предотвратить переполнение и конфликтные ситуации в домене .us. Доменное имя добавляется в конце имени хост-компьютера и формирует уникальное имя в сети Internet для данного компьютера. Это и есть известный сегодня формат имени хоста, с которым вы, вероятно, уже сталкивались не один раз. В [табл. 4.1](https://intuit.ru/studies/professional_skill_improvements/1274/courses/116/lecture/3361?page=1#table.4.1) дается описание некоторых доменов верхнего уровня.

|  |
| --- |
| Таблица 4.1. Доменные имена верхнего уровня |
| **Имя домена** | **Описание** |
| .com | Коммерческие организации в США |
| .edu | Образовательные учереждения в США |
| .gov | Государственные органы США |
| .mil | Министерство обороны США |
| .net | Провайдеры сети Internet в США |
| .org | Общественные организации в США, не имеющие целью получение прибыли |
| .us | Другие организации в США |
| .ca | Организации в Канаде |
| .de | Организации в Германии |
| (другие коды стран) | Организации в других странах |

По мере дальнейшего роста сети Internet все домены верхнего уровня были поделены на поддомены или зоны. Каждая зона представляет собой независимый домен, но при обращении к базе данных имен запрашивает родительский домен. Родительская зона гарантирует дочерней зоне право на существование и отвечает за ее поведение в сети (точно так же, как и в реальной жизни). Каждая зона должна иметь по крайней мере два сервера DNS, которые поддерживают базу данных DNS для этой зоны.

Основные условия для работы серверов DNS одной зоны — наличие отдельного соединения с сетью Internet и размещение их в различных сетях для обеспечения отказоустойчивости. Поэтому многие организации полагаются на провайдеров Internet, которые ведут в их интересах вторичные и третичные серверы DNS.

Уникальный адрес в сети Internet для данной зоны формируется добавлением к имени хоста доменного имени. Таким образом, компьютер fred в домене smallorg.org будет называться fred.smallorg.org. Постарайтесь не запутаться, так как домен может содержать как хосты, так и зоны. Например, домен smallorg.org может содержать хост fred.smallorg.org и в то же время вести зону acctg.smallorg.org, которая является поддоменом и может содержать еще один хост barney.acctg.smallorg.org. Хотя это и упрощает базу данных имен, однако делает поиск хостов в сети Internet более сложным. На [рис. 4.2](https://intuit.ru/studies/professional_skill_improvements/1274/courses/116/lecture/3361?page=2#image.4.2) показан пример домена и связанного с ним поддомена.



**Рис. 4.2.**Пример домена и поддомена в сети Internet

**Доменные имена в сети Internet**

За последние несколько лет доменные имена в Internet стали горячей темой. В прошлом управление и распределение доменов по США вида .com, .net и .org находилось в ведении одной корпорации — Internic Corporation. Однако недавно для упорядочения процесса выделения доменов была учреждена специальная бесприбыльная организация — Internet Corporation for Assigned Names and Numbers (*ICANN*). Теперь на территории США только этой организации принадлежит право в распределении доменных имен. Теперь оплата доменного имени может производиться несколькими организациями. Прежде чем использоваться на территории США, все доменные имена должны быть согласованы с *ICANN*. В Европе этими вопросами занимается организация RIPE (Reneaux IP Europe) — Прим. ред.).

В системе DNS реализуются три сценария поиска IP-адреса в базе данных.

* Компьютер, которому необходимо получить соединение с другим компьютером в той же зоне, посылает запрос локальному DNS-серверу зоны на поиск IP-адреса удаленного компьютера. Локальный DNS-сервер, имеющий этот адрес в локальной базе данных имен, возвращает запрашиваемый IP-адрес компьютеру, который посылал запрос.
* Компьютер, которому необходимо получить соединение с компьютером в другой зоне запрашивает локальный DNS-сервер своей зоны. Локальный DNS-сервер обнаруживает, что нужный компьютер находится в другой зоне, и формирует запрос корневому DNS-серверу. Корневой DNS-сервер спускается по дереву серверов DNS и находит соответствующий локальный DNS-сервер. От него он получает IP-адрес запрашиваемого компьютера. Затем корневой DNS-сервер передает этот адрес локальному серверу DNS, который послал запрос. Локальный DNS-сервер возвращает IP-адрес компьютеру, с которого был подан запрос. Совместно с IP-адресом передается специальное значение — время жизни TTL (time to live). Это значение указывает локальному DNS-серверу, сколько времени он может хранить IP-адрес удаленного компьютера у себя в кэше. Благодаря этому увеличивается скорость обработки последующих запросов.
* Компьютер, которому необходимо повторно получить соединение с компьютером в другой зоне запрашивает локальный DNS-сервер своей зоны. Локальный DNS-сервер проверяет, нет ли этого имени в его кэше и не истекло ли еще значение TTL. Если адрес еще в кэше и значение TTL не истекло, то IP-адрес посылается запрашивающему компьютеру. Это считается неавторизованным ответом, так как локальный DNS-сервер считает, что с момента последнего запроса IP-адрес удаленного компьютера не изменился

Во всех трех случаях компьютеру для поиска какого-либо компьютера в сети Internet нужен лишь IP-адрес локального сервера DNS. Дальнейшую работу по поиску IP-адреса, соответствующего запрошенному имени, выполняет локальный DNS-сервер. Как видите, теперь все намного проще для локального компьютера. На [рис. 4.3](https://intuit.ru/studies/professional_skill_improvements/1274/courses/116/lecture/3361?page=2#image.4.3) показан механизм работы системы DNS.



**Рис. 4.3.**Методы преобразования в системе DNS

По мере роста дерева DNS, к серверам системы доменных имен предъявлялись новые требования. Как уже упоминалось ранее, родительские DNS-серверы должны иметь IP-адреса своих дочерних серверов DNS, чтобы правильно обрабатывать DNS-запросы на преобразование имен в IP-адреса. Чтобы DNS-запросы обрабатывались правильно, поиск по дереву DNS должен начинаться из какой-то определенной точки. В период младенчества сети Internet большинство запросов на поиск имен приходилось на локальные имена хостов. Основная часть DNS-трафика проходила внутри локальной зоны и лишь в худшем случае достигала родительских серверов DNS. Однако с ростом популярности Internet и, в частности Web, все больше DNS-запросов формировалось к удаленным хостам вне локальной зоны. Когда DNS-сервер не находил имя хоста в своей базе данных, он вынужден был запрашивать удаленный DNS-сервер. Наиболее подходящими кандидатами для удаленных DNS-серверов, естественно, стали серверы DNS верхнего уровня, которые обладают полной информацией о дереве доменов и способны найти нужный DNS-сервер, ответственный за зону, к которой принадлежит запрашиваемый хост. Затем они же возвращают IP-адрес нужного хоста локальному DNS-серверу. Все это приводит к колоссальным перегрузкам корневых серверов системы DNS. К счастью, их не так много и все они равномерно распределяют нагрузку между собой. Локальные DNS-серверы работают с серверами DNS доменов верхнего уровня с помощью протокола DNS, который рассматривается далее в этой лекции.

Система DNS — улица с двусторонним движением. DNS не только отыскивает IP-адрес по заданному имени хоста, но способна выполнять и обратную операцию, т.е. по IP-адресу определять имя хоста в сети. Многие Web- и FTP-серверы в сети Internet ограничивают доступ на основе домена, к которому принадлежит обратившийся к ним клиент. Получив от клиента запрос на установку соединения, сервер передает IP-адрес клиента DNS-серверу как обратный DNS-запрос. Если клиентская зона DNS настроена правильно, то на запрос будет возвращено имя клиентского хоста, на основе которого затем принимается решение о том, допустить данного клиента на сервер или нет.

**Протокол DHCP и DNS**

Если у вас используется протокол динамического конфигурирования хостов DHCP (Dynamic Host Configuration Protocol), с помощью которого производится автоматическое назначение IP-адресов рабочим станциям, то, возможно, понадобится создать записи в базе данных DNS для всего диапазона адресов, которые могут назначаться сервером согласно протоколу DHCP. Часто каждому адресу из динамического диапазона может назначаться одно и то же имя. Например, station1.smallorg.org.

#### Записи в базе данных DNS

DNS-сервер, отвечающий за имена хостов в своей зоне, должен хранить информацию о хостах в базе данных и выдавать ее по запросу с удаленных компьютеров. База данных DNS представляет собой текстовый файл, состоящий из исходных записей RR (resource records). Эти записи описывают компьютеры и их функции в локальной зоне. Для организации обмена информацией с удаленными серверами DNS на сервере Linux должно быть запущено программное обеспечение сервера DNS (обычно это программа named ). Чуть позже в этой лекции мы рассмотрим работу этой программы.

Прежде всего в базе данных сервера DNS должна быть объявлена зона, за которую данный сервер несет ответственность. Далее в ней должны быть объявлены все хост-компьютеры, имеющиеся в зоне. И, наконец, в базе данных можно объявлять специальную информацию, касающуюся зоны (например, о серверах электронной почты и DNS-серверах). Формат записи базы данных был разработан таким образом, чтобы DNS-сервер мог почерпнуть из нее любую информацию, нужную для его работы. В [табл. 4.2](https://intuit.ru/studies/professional_skill_improvements/1274/courses/116/lecture/3361?page=3#table.4.2) приведены основные типы исходных записей, которые могут присутствовать в базе данных DNS. База данных DNS в последнее время стала темой для дискуссий среди исследователей. Так как многие хотят дополнить ее новыми возможностями и наряду с этим повысить уровень безопасности. В настоящее время в базу данных DNS постоянно вносятся новые типы записей. В [табл. 4.2](https://intuit.ru/studies/professional_skill_improvements/1274/courses/116/lecture/3361?page=3#table.4.2) отражены лишь основные типы записей, которые необходимы для открытия и ведения новой зоны в базе данных DNS.

|  |
| --- |
| Таблица 4.2. Типы записей в базе данных DNS |
| **Тип записи** | **Описание** |
| SOA | Начало полномочий |
| A | Internet-адрес |
| NS | Сервер имен |
| CNAME | Каноническое имя (имя машины) |
| HINFO | Информация о машине (хосте) |
| MX | Почтовый сервер |
| PTR | Указатель |

Каждый DNS-сервер домена должен содержать исходные записи для всех хостов этого домена. В нем должна присутствовать одна запись SOA, которая вносится в самом начале базы данных. Все остальные исходные записи могут добавляться далее в произвольном порядке. На [рис. 4.4](https://intuit.ru/studies/professional_skill_improvements/1274/courses/116/lecture/3361?page=3#image.4.4) показан пример базы данных DNS для сети с [рис. 4.2](https://intuit.ru/studies/professional_skill_improvements/1274/courses/116/lecture/3361?page=2#image.4.2). В следующем разделе основные типы записей будут рассмотрены более подробно.



**Рис. 4.4.**Пример ведения записей DNS для небольшой сети

##### Запись "начало полномочий" (SOA- Start of Authority)

Всякая база данных DNS должна начинаться с записи SOA, которая отмечает начало зоны, описываемой в базе данных.

Формат записи SOA:

domain name [TTL] [class] SOA origin

person (

serial number

refresh

retry

expire

minimum)

Здесь domain name — имя зоны, которая далее описывается в базе данных (можно использовать также знак @ для обозначения домена по умолчанию для данного компьютера).

TTL — время (в секундах), в течение которого запрашивающий компьютер может хранить любую информацию в своем *локальном кэше* DNS. Этот параметр не является обязательным.

class обозначает протокол, который используется в системе (в нашем случае везде будет присутствовать класс IN для сети Internet ). Этот параметр также необязателен и по умолчанию ставится в значение IN.

origin — имя компьютера, где ведется основная зона. Будьте осторожны при расстановке разделительных точек ( .)! Не забудьте поставить точку после имени хоста, иначе доменное имя будет добавлено к имени хоста (если конечно вы не хотите обратного).

person — адрес электронной почты лица, ответственного за данную зону. Здесь имеется небольшое отличие от предыдущих версий DNS. Так как значок @ уже применяется для указания домена по умолчанию, его нельзя еще раз указывать в адресе электронной почты. Вместо знака @ используйте обычную точку " .". Так, например, вместо адреса sysadm@smallorg.org используйте sysadm.smallorg.org. Если в адресе электронной почты имеются другие точки, то вместо них используйте обратную косую черту \. Например, адрес john.jones@smallorg.org будет преобразован в john\jones.smallorg.org.

serial number является уникальным числом, которое идентифицирует версию файла базы данных. Очень часто для этой цели используется дата создания файла или последней его модификации и номер версии файла (например, 200026061).

refresh — интервал времени (в секундах), по истечении которого вторичный DNS-сервер должен опрашивать первичный DNS-сервер о текущей версии записи SOA. Если версия отличается от имеющейся во вторичном сервере DNS, то он пошлет запрос на обновление своей базы данных. Обычно для этого параметра устанавливается значение 1 час (3600 секунд).

retry — интервал времени (в секундах), по истечении которого вторичный DNS-сервер повторяет попытку обновить содержимое своей базы после неудачной предыдущей попытки.

expire — это интервал времени (в секундах) по истечении которого вторичный DNS-сервер может использовать данные, полученные от первичного DNS-сервера, без обновлений. Обычно это значение довольно большое, например 3600000 секунд (около 42 дней).

minimum — интервал времени (в секундах), который должен использоваться как TTL для всех исходных записей в зоне. Обычно 86400 секунд (т.е. одного дня) достаточно для этого параметра.

##### Запись Internet-адреса (A)

Каждый хост в зоне должен иметь в базе данных запись типа А, которая определяет его имя в сети Internet. Формат записи типа А:

host [TTL] [ class ] A address,

где host — полностью определенное имя компьютера (включая доменное имя), а address — IP-адрес этого компьютера.

##### Каноническое имя (CNAME)

Кроме обычного имени, некоторые хосты могут иметь два-три дополнительных имени (псевдонима). Это удобно, когда требуется идентифицировать различные сервисы. При этом нет необходимости переименовывать компьютеры в домене. Например, имя, зарезервированное для Web-сервера, — www.smallorg.org. Запись CNAME связывает псевдонимы с реальными именами хостов. Ее формат следующий:

nickname [TTL] [ class ] CNAME host name

##### Запись сервера имен (NS)

В каждой зоне должно быть по крайней мере два сервера DNS. Записи NS служат для их идентификации другими DNS-серверами, которые пытаются преобразовать имена хостов, относящихся к данной зоне. Формат записи NS:

domain [TTL] [ class ] NS server

domain — имя домена в зоне, за которую отвечает DNS-сервер. Если оно опущено, то NS -запись относится к зоне, указанной в записи SOA.

server — имя DNS-сервера. Для него также должна существовать соответствующая запись типа А, где будет указан его IP-адрес.

##### Запись с информацией о хосте (HINFO)

Дополнительную информацию о параметрах компьютера можно сделать доступной для DNS посредством объявления записи HINFO. Формат записи HINFO следующий:

host [TTL] [ class ] HINFO hardware software

host — имя хост-компьютера, к которому относится информация из записи.

hardware — краткая характеристика аппаратного обеспечения компьютера.

software — тип и номер версии операционной системы, используемой в компьютере.

##### Запись указателя (PTR)

Кроме записи А, для каждого компьютера в зоне создается также запись типа PTR. Это позволяет серверу DNS выполнять обратное преобразование для IP-адресов. Без этой информации удаленные серверы не смогут определить имя домена, в котором находится ваш компьютер. Формат записи PTR:

IN-ADDR name [TTL] [ class ] PTR name

IN-ADDR name — это обратное отражение имени DNS в IP-адрес. Это, может показаться немного запутанным, но соответствует действительности. Это имя позволяет DNS-серверу работать в обратном порядке, когда преобразованию в имя подлежит IP-адрес компьютера. Адрес IN-ADDR.ARPA является специальным доменом, который поддерживает работу шлюза и преобразование Internet-адресов в имена хостов. При этом нет необходимости в обратных запросах, так как IP-адрес соответствует фиктивному имени хоста. Так, например, имя IN-ADDR name для компьютера с IP-адресом 192.168.0.1 будет 1.0.168.192.IN-ADDR.ARPA.

name в конце записи обозначает имя хост-компьютера согласно записи А.

##### Запись почтового сервера (MX)

Наиболее важными (по крайней мере для администраторов систем электронной почты) являются записи MX. С их помощью удаленные серверы электронной почты узнают, куда отправлять почту для вашей зоны. Формат записи типа МХ следующий:

name [TTL] [ class ] MX preference host

Здесь name — имя зоны (если оно опущено, то запись относится к зоне, указанной в SOA ). Можно также указывать здесь имя хоста в сети, на который вы хотели бы перенаправлять всю электронную почту.

Параметр preference — целое число, обозначающее порядок подключения, если почта принимается от нескольких почтовых серверов. Он представляет собой приоритет, на основе которого, принимается решение о предпочтительности того или иного почтового сервера. Так, значение 0 соответствует наивысшему приоритету среди почтовых серверов. Приоритет убывает с увеличением значения preference. С помощью этого параметра обычно создаются первичные и вторичные почтовые серверы для данной зоны. Когда от удаленного почтового сервера на DNS-сервер поступает запрос о почтовом сервере, ответственном за данную зону, то в ответ ему посылается список всех почтовых серверов с соответствующими им приоритетами. Удаленный почтовый сервер попытается соединиться с сервером, у которого самый высокий приоритет из списка, и, если попытка оказалась неудачной, продолжает устанавливать соединение с менее приоритетными серверами.

host — имя почтового сервера. Ему также должна соответствовать запись типа А, где будет указан его адрес IP.

#### Пример базы данных DNS для домена

Если вы решили не обременять себя ведением домена и почтового сервера, а отдать эти функции на откуп провайдеру Internet, то он организует поддержку соответствующих записей в своей базе DNS, представляя ваш домен в сети Internet. Запись SOA будет идентифицировать ваш домен, но указывать на хост провайдера, который будет являться авторизованным (уполномоченным) хостом. Записи NS для вашего домена также будут указывать на DNS-серверы провайдера, а в МХ -записях будут указаны почтовые серверы провайдера. Для всей остальной сети Internet в этом случае эти компьютеры являются частью вашей сети, даже если реально они к ней не относятся. В листинге 4.1 представлен пример описания зоны в базе данных DNS провайдера Internet.

1 smallorg.org IN SOA master.isp.net. postmaster.master.isp.net (

2 2000260601 ;unique serial number

3 8H ; refresh rate

4 2H ;retry period

5 1W ; expiration period

6 1D) ; minimum

7

8 NS ns1.isp.net. ;defines primary name server

9 NS ns2.isp.net. ; defines secondary name server

10

11 MX 10 mail1.isp.net. ; defines primary mail server

12 MX 20 mail2.isp.net. ; defines secondary mail server

13

14 www CNAME host1.isp.net. ;defines your www server at the ISP

15 ftp CNAME host1.isp.net ; defines your FTP server at the ISP

16

17 host1.isp.net A 10.0.0.1

18

19 1.0.0.10.IN-ADDR.ARPA PTR host1.isp.net ; pointer address for reverse DNS

Листинг 4.1. Описание зоны в базе DNS

В строках 1–6 показана запись SOA для вашего нового домена. Здесь провайдер указывает, что ваше доменное имя smallorg.org ведется его сервером master.isp.net. В строках 8 и 9 определяются первичный и вторичный серверы DNS, которые будут использоваться для преобразования имен ваших хост-компьютеров (заметьте, принадлежащих провайдеру). Строки 11 и 12 определяют первичный ( mail1.isp.net ) и вторичный ( mail2.isp.net ) почтовые серверы, которые будут принимать и накапливать электронную почту для вашего домена. Имя хоста www.smallorg.org — это псевдоним, указывающий на ваш Web-сервер. Адрес ftp.smallorg.org указывает на ваш FTP-сервер, который также находится на хосте провайдера. Все это элементарные услуги, предоставляемые провайдерами сети Internet тем клиентам, которые не могут позволить себе постоянное соединение с сетью Internet по выделенной линии, но желают предоставить своим клиентам Web- и FTP-сервисы. Строки 17 и 19 задают информацию об IP-адресе в сети. С ее помощью удаленные клиенты могут найти и подключиться к этому серверу через Internet. Довольно часто записи типа PTR (см. строку 19) выделяются в отдельную базу данных. Это делается с целью упрощения баз данных системы DNS. В приведенном нами примере это непринципиально, так как мы имеем дело лишь с одной записью PTR, однако часто можно встретить базы данных с дюжиной и более записей PTR.

Теперь, когда сервер DNS имеет правильно сконфигурированную базу данных, пришло время обмениваться запросами на преобразование имен хостов с другими DNS-серверами. Этот обмен DNS-запросами производится согласно специально разработанному протоколу DNS.

#### Протокол DNS

Протокол DNS выполняет две основные функции. Он позволяет клиентским компьютерам запрашивать DNS-сервер об IP-адресе или имени какого-либо хоста в сети, а также позволяет производить обмен информацией между базами данных серверов DNS. В этом протоколе используется стандартный формат типа "запрос-ответ", где клиент посылает пакет запроса, и сервер отвечает либо пакетом с информацией, полученной из базы данных, либо сообщением об ошибке, в котором указывается причина отказа в обработке запроса. В своей работе этот протокол использует порт 53 и хорошо известные протоколы — TCP или UDP. Причем в последнее время UDP стал более распространенным методом транспортировки пакетов по сети Internet. Пакет DNS состоит из пяти полей: заголовка, вопроса, ответа, полномочий и поля дополнительной информации. На [рис. 4.5](https://intuit.ru/studies/professional_skill_improvements/1274/courses/116/lecture/3361?page=5#image.4.5) показана общая структура пакета DNS.



**Рис. 4.5.**Описание пакета протокола DNS

##### Поле заголовка

В поле заголовка содержится информация о пакете и его назначении. В нем дается общее описание пакета (пакет запроса или пакет ответа) и указывается количество данных, содержащихся в каждом поле данных пакета. *Описание заголовка* приводится в [табл. 4.3](https://intuit.ru/studies/professional_skill_improvements/1274/courses/116/lecture/3361?page=5#table.4.3).

|  |
| --- |
| Таблица 4.3. Поле заголовка DNS-пакета |
| **Бит** | **Описание** |
| 0-15 | ID |
| 16 | QR |
| 17-20 | *OPCODE* |
| 21 | AA |
| 22 | TC |
| 23 | RD |
| 24 | RA |
| 25-27 | Z |
| 28-31 | RCODE |
| 32-47 | QDCOUNT |
| 48-63 | ANCOUNT |
| 64-79 | NSCOUNT |
| 80-95 | ARCOUNT |

Биты ID являются уникальным 16-битовым идентификационным номером пакета запроса. Пакет ответа, формируемый сервером, также использует этот идентификационный номер, чтобы клиент мог сопоставить ответ сервера со своим запросом. Бит QR обозначает тип пакета (пакет запроса — 0, пакет ответа — 1). Поле *OPCODE* определяет тип запроса — стандартный (0), обратный (1) или запрос о статусе сервера (2).

Следующие четыре бита определяют различные параметры пакета. Бит AA устанавливается, когда ответ является авторитетным (данные поступают напрямую от DNS-сервера, ответственного за зону). Неавторитетные ответы могут поступать от серверов DNS, в кэше которых сохранилась информация об исходных записях от предыдущих запросов. Эта информация считается неавторитетной, так как есть вероятность, что с момента последнего обращения к серверу информация была изменена. Бит TC устанавливается, когда требуется урезать данные в пакете до вида, удобного для передачи по сети. Такое вполне возможно при использовании протокола UDP, согласно которому размер пакета не должен превышать 512 байт. Бит RD включается, когда клиент желает рекурсивно запрашивать DNS-сервер на постоянной основе. Если этот бит установлен, то DNS-сервер будет запрашивать другие DNS-серверы, пока не получит ответ. Если этот бит не установлен, то DNS-сервер будет возвращать на запрос любую информацию, которая у него имеется. Бит RA устанавливается, чтобы уведомить клиента о возможности *рекурсивного запроса* на данный сервер. Биты Z в настоящее время не используются и зарезервированы на будущее.

Биты RCODE используются только в пакетах ответов. Они отображают состояние ответа — без ошибок (0), ошибки в пакете запроса (1), внутренние ошибки не дали возможности серверу обработать запрос (2), имя, указанное в запросе, не существует (3), данный тип запроса не поддерживается сервером (4) и сервер отказался обработать запрос (5).

Остальные четыре параметра заголовка представляют собой 16-битовые числа и используются в качестве счетчиков. С их помощью ведется учет количества исходных записей, возвращаемых в пакете. QDCOUNT отображает количество запросов (в пакет может включаться более одного запроса). ANCOUNT — количество исходных записей, включенных в ответ. NSCOUNT обозначает число исходных записей об авторитетных серверах имен, а ARCOUNT — число записей в поле дополнительной информации.

##### Поле вопроса

Поле вопроса содержит запросы, ответы на которые клиент желает получить от DNS-сервера. В одном пакете DNS может содержаться несколько запросов. Количество запросов в пакете определяется параметром QDCOUNT из поля заголовка. Поле вопроса состоит из трех частей: списка преобразуемых доменных имен; поля типов записей, которые клиент желает получить в ответе, и параметра класса запроса. Список преобразуемых доменных имен представляет собой список имен, для которых клиент желает получить IP-адреса. Для формирования списка имен используется специальный формат. Перед каждым именем ставится однобайтное значение, которое определяет длину имени. Конец списка обозначается именем с нулевой длиной. После текстовой части следует двубайтная запись QTYPE. В ней определяется, в каком виде клиент желает принимать информацию о имеющихся доменах. Эти значения полностью соответствуют типам исходных записей в DNS. Например, для того чтобы найти почтовый сервер для определенного домена, вам следует воспользоваться типом записи МХ. И, наконец, последний параметр в поле вопроса — QCLASS. Он определяет класс запроса, который в нашем случае для сети Internet всегда будет IN.