Лабораторная работа № 9

Служба доменных имен (DNS)

Краткие теоретические сведения

1. Основы DNS.

Система доменных имен (DNS – Domain Name System) – это распределенная база данных, которая используется приложениями TCP/IP, для установления соответствия между именами узлов и IP адресами. DNS также используется для маршрутизации электронной почты. Термин распределенная, потому что на одном узле Internet не хранится вся необходимая информация. Каждый узел (университет, университетский городок, компания или отдел внутри компании) поддерживает собственную информационную базу данных и запускает программу сервер, которая может отправить запрос по Internet к другим системам. DNS предоставляет протокол, который позволяет клиентам и серверам общаться друг с другом.

С точки зрения приложения, доступ к DNS осуществляется посредством определителя (resolver) (определитель – подпрограммы, которые используются для создания, отправки и интерпретации пакетов, используемых серверами имен Internet). В Unix системах, к разборщику можно получить доступ через две библиотечные функции, gethostbyname(3) и gethostbyaddr(3), которые линкуются с приложением, когда оно строится. Первая воспринимает в качестве аргумента имя узла и возвращает IP адрес, а вторая воспринимает в качестве аргумента IP адрес и возвращает имя узла. Разборщик устанавливает контакты с одним или несколькими серверами DNS (name servers), чтобы установить это соответствие.

Определитель – это часть приложения. Он не является частью ядра операционной системы как протоколы TCP/IP. Приложение должно конвертировать имя узла в IP адрес, перед тем как оно попросит TCP открыть соединение или послать датаграмму с использованием UDP. Протоколы TCP/IP внутри ядра ничего не знают о DNS.

RFC 1034 [Mockapetris 1987a] описывает концепции, лежащие в основе DNS, а RFC 1035 [Mockapetris 1987b] содержит подробности разработки и спецификации DNS. Наиболее широкоиспользуемая реализация DNS, как разборщика, так и сервера – BIND (Berkeley Internet Name Domain). Процесс сервера называется named.

Пространство имен DNS имеет иерархическую структуру, которая внешне напоминает файловую систему Unix. На рисунке 1 показана иерархическая организация DNS.

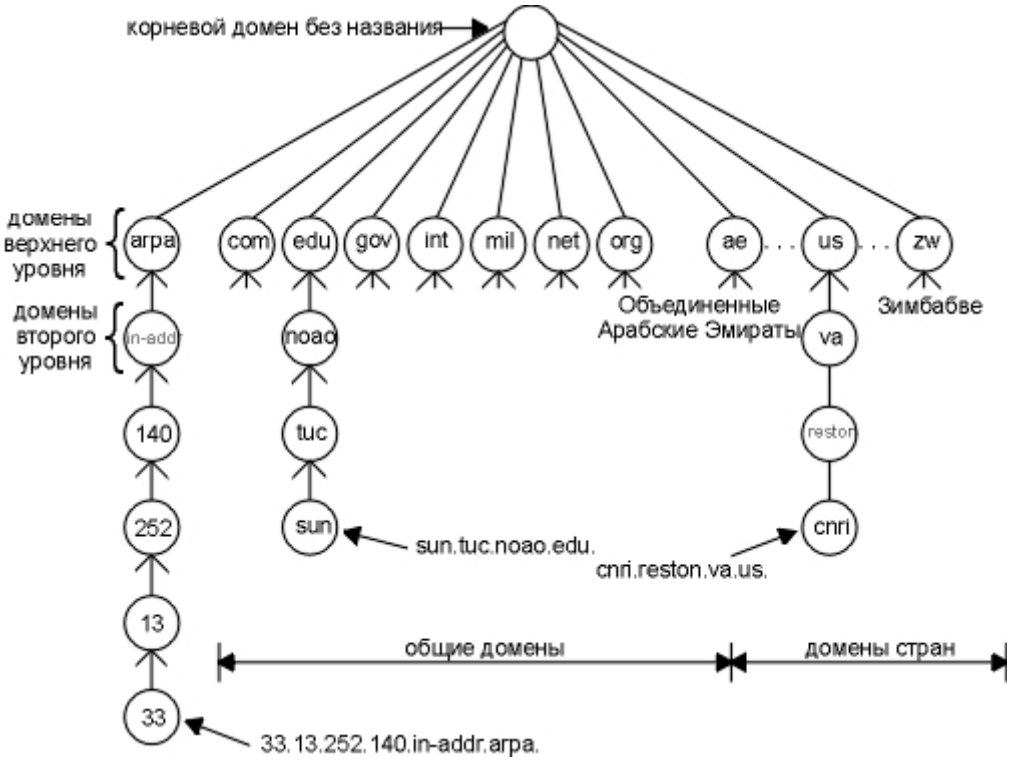

Рисунок 1

Каждый узел (кружочки на рисунке 1) имеет метку длиной до 63 символов. Корень дерева это специальный узел без метки. Метки могут содержать заглавные буквы или маленькие. Имя домена (domain name) для любого узла в дереве

– это последовательность меток, которая начинается с узла выступающего в роли корня, при этом метки разделяются точками. (Здесь видно отличие от файловой системы Unix, где полный путь всегда начинается с вершины (корня) и опускается вниз по дереву.) Каждый узел дерева должен иметь уникальное имя домена, однако одинаковые метки могут быть использованы в различных точках дерева.

Имя домена, которое заканчивается точкой, называется абсолютным именем домена (absolute domain name) или полным именем домена (FQDN – fully qualified domain name). Например, sun.tuc.noao.edu. . Если имя домена не заканчивается на точку, подразумевается, что имя должно быть завершено. Как будет закончено имя, зависит от используемого программного обеспечения DNS. Если незаконченное имя состоит из двух или более меток, его можно воспринимать как законченное или полное; иначе справа от имени должен быть добавлен локальный суффикс. Например, имя comsys может быть завершено локальным суффиксом .tuc.noao.edu. .

Домены верхнего уровня поделены на три зоны:

1. arpa это специальный домен, используемый для сопоставления адрес – имя.

2. Семь 3-символьных доменов называются общими (generic) доменами. В некоторых публикациях они называются организационными (organizational) доменами.

3. Все 2-символьные домены, основанные на кодах стран, можно найти в ISO 3166. Они называются доменами стран (country), или географическими (geographical) доменами.

На рисунке 2 приведен список обычной классификации семи основных доменов.

Домен	Описание
com	коммерческие организации
edu	учебные организации
gov	правительственные организации США
int	международные организации
mil	военные организации США
net	сети
org	другие организации

Рисунок 2

Одна важная характеристика DNS, не показанная на рисунке 1, это передача ответственности внутри DNS. Не существует организации, которая бы управляла и обслуживала все дерево в целом и каждую метку в отдельности. Вместо этого, одна организация (NIC) обслуживает только часть дерева (домены верхнего уровня), а ответственность за определенные зоны передает другим организациям.

Зона (zone) это отдельно администрируемая часть дерева DNS. Например, домен второго уровня noao.edu это отдельная зона. Многие домены второго уровня поделены на меньшие зоны. Например, университет может поделить свою зону на подзоны по факультетам, а компания может поделить себя на зоны по принципу деления на филиалы или отделы.

С того момента, как выбрана организация или персона, которая несет ответственность за управление зоной, эта организация или персона должна организовать несколько серверов DNS (name servers) для этой зоны. Как только в зоне появляется новая система, администратор этой зоны помещает имя и IP адрес нового узла в базу данных сервера DNS.

Сервер DNS, скажем, обслуживает одну зону или несколько зон. Человек, который несет ответственность за зону, администрирует основной сервер DNS (primary name server) для этой зоны и один или несколько вторичных серверов DNS (secondary name servers). Первичный и вторичный сервера должны быть независимы и избыточны таким образом, чтобы система DNS не вышла из строя при отказе одного из серверов.

Основное отличие между первичными и вторичными серверами заключается в том, что первичные загружают всю необходимую информацию из дисковых файлов, тогда как вторичные получают информацию от первичного. Процесс передачи информации от первичного сервера вторичному называется передачей зоны (zone transfer). Когда в зоне появляется новый узел, администратор добавляет соответствующую информацию (минимум, имя и IP адрес) в дисковый файл на первичном сервере. После чего первичный сервер DNS уведомляется о необходимости повторно считать свои конфигурационные файлы. Вторичные сервера регулярно опрашивают первичные (обычно каждые 3 часа), и если первичные содержат новую информацию, вторичный получает ее с использованием передачи зоны.

Что произойдет, если сервер DNS не содержит необходимой информации? Он должен установить контакт с другим сервером DNS. (В этом заключается распределенная природа DNS.) Однако не каждый сервер DNS знает, как обратиться к другому серверу. Вместо этого каждый сервер DNS должен знать, как установить контакт с корневыми серверами DNS (root name servers). В апреле 1993 года существовало восемь корневых серверов, все первичные сервера должны знать IP адреса каждого корневого сервера. (Эти IP адреса находятся в конфигурационных файлах первичного сервера. Первичные сервера должны знать именно IP адреса корневых серверов, а не их DNS имена.) Корневой сервер, в свою очередь, знает имена и положения (IP адрес) каждого официального сервера DNS для всех доменов второго уровня. При

этом возникает последовательный процесс: запрашивающий сервер должен установить контакт с корневым сервером. Корневой сервер сообщает запрашивающему серверу о необходимости обратиться к другому серверу и так далее.

Фундаментальная характеристика DNS – это кэширование (caching). Когда DNS сервер получает информацию о соответствии (например, IP адресов именам узлов), он кэширует эту информацию таким образом, что в случае следующего запроса может быть использована информация из кэша, дополнительный запрос на другие сервера не делается.

2. Формат сообщений DNS.

Для DNS запроса и для DNS отклика используется одинаковый формат. На рисунке 3 показан общий формат DNS сообщения.

Рисунок 3

Сообщение содержит фиксированный 12-байтный заголовок, за которым следуют четыре поля переменной длины.

Значение в поле идентификации (identification) устанавливается клиентом и возвращается сервером. Это поле позволяет клиенту определить, на какой запрос пришел отклик.

16-битовое поле флагов (flags) поделено на несколько частей, как показано на рисунке 4.

где,
QR - тип сообщения
opcode - код операции
AA - авторитетный ответ
TC - "обрезано"
RD - "требуется рекурсия"
RA - "рекурсия возможна"
rcode - код возврата

Рисунок 4

- QR (тип сообщения), 1-битовое поле: 0 обозначает – запрос, 1 обозначает – отклик.
- opcode (код операции), 4-битовое поле. Обычное значение 0 (стандартный запрос). Другие значения – это 1 (инверсный запрос) и 2 (запрос статуса сервера).
- AA – 1-битовый флаг, который означает "авторитетный ответ" (authoritative answer). Сервер DNS имеет полномочия для этого домена в разделе вопросов.

• TC – 1-битовое поле, которое означает "обрезано" (truncated). В случае UDP это означает, что полный размер отклика превысил 512 байт, однако были возвращены только первые 512 байт отклика.

• RD – 1-битовое поле, которое означает "требуется рекурсия" (recursion desired). Бит может быть установлен в запросе и затем возвращен в отклике. Этот флаг требует от DNS сервера обработать этот запрос самому (т.е. сервер должен сам определить требуемый IP адрес, а не возвращать адрес другого DNS сервера), что называется рекурсивным запросом (recursive query). Если этот бит не установлен и запрашиваемый сервер DNS не имеет авторитетного ответа, запрашиваемый сервер возвратит список других серверов DNS, к которым необходимо обратиться, чтобы получить ответ. Это называется повторяющимся запросом (iterative query).

• RA – 1-битовое поле, которое означает "рекурсия возможна" (recursion available). Этот бит устанавливается в 1 в отклике, если сервер поддерживает рекурсию. Большинство серверов DNS поддерживают рекурсию, за исключением нескольких корневых серверов (коневые сервера не в состоянии обрабатывать рекурсивные запросы из-за своей загруженности).

• Это 3-битовое поле должно быть равно 0.

• rcode это 4-битовое поле кода возврата. Обычные значения: 0 (нет ошибок) и 3 (ошибка имени). Ошибка имени возвращается только от полномочного сервера DNS и означает, что имя домена, указанного в запросе, не существует.

Следующие четыре 16-битных поля указывают на количество пунктов в четырех полях переменной длины, которые завершают запись. В запросе количество вопросов (number of questions) обычно равно 1, а остальные три счетчика равны 0. В отклике количество ответов (number of answers) по меньшей мере равно 1, а оставшиеся два счетчика могут быть как нулевыми, так и ненулевыми.

Формат каждого вопроса в разделе вопросов (question) показан на рисунке 5. Обычно присутствует только один вопрос.

Имя запроса (query name) это искомое имя. Оно выглядит как последовательность из одной или нескольких меток. Каждая метка начинается с 1-байтового счетчика, который содержит количество следующих за ним байт. Имя заканчивается байтом равным 0, который является меткой с нулевой длиной. И является, в свою очередь, меткой корня. Каждый счетчик байтов должен быть в диапазоне от 0 до 63, так как длина метки ограничена 63 байтами.

Рисунок 5

В отличие от многих других форматов сообщений, этому полю разрешено заканчиваться на ограничителе не равном 32 битам. Заполнение не используется.

На рисунке 6 показано, как хранится имя домена gemini.tuc.noao.edu.

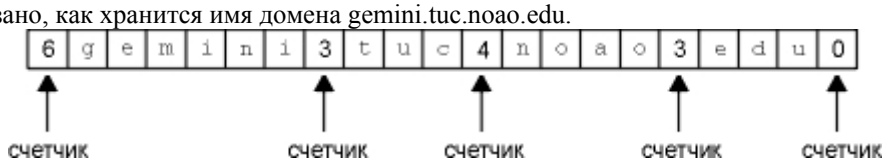

Рисунок 6

У каждого вопроса есть тип запроса (query type), а каждый отклик (называемый записью ресурса) имеет тип (type). Существует около 20 различных значений, некоторые из которых в настоящее время уже устарели. На рисунке 7 показаны некоторые из этих значений. Тип запроса это надмножество (множество, подмножеством которого является данное множество) типов: два из показанных значений, могут быть использованы только в вопросах.

Имя	Цифровое значение	Описание	тип (type)?	тип запроса (query type)?
A	1	IP адрес	Да	Да
NS	2	сервер DNS	Да	Да
CNAME	5	каноническое имя	Да	Да
PTR	12	запись указателя	Да	Да
HINFO	13	информация о узле	Да	Да

MX	15	запись об обмене почтой	Да	Да
AXFR	252	запрос на передачу зоны	Нет	Да
* или ANY	255	запрос всех записей	Нет	Да

Рисунок 7

Наиболее распространенный тип запроса – тип A, который обозначает, что необходим IP адрес для запрашиваемого имени (query name). PTR запрос требует имена, соответствующие IP адресу.

Класс запроса (query class) обычно равен 1, что указывает на адреса Internet. (В некоторых случаях поддерживаются не-IP значения.)

Последние три поля в DNS сообщении это ответы (answers), полномочия (authority) и дополнительная информация (additional information), общий формат называется записью ресурса (RR – resource record). На рисунке 8 показан формат записи ресурса.

Рисунок 8

Имя домена (domain name) это имя, которому соответствуют следующие данные ресурса. Формат имени домена тот же, что описан ранее для поля имени запроса (query name) (рисунок 6).

Тип (type) указывает на один из типов кодов RR. Это то же самое, что и значения типа запроса (query type). Для данных Internet класс (class) обычно установлен в 1.

Поле время жизни (time-to-live) это количество секунд, в течение которых RR может быть кэширована клиентом. Обычно TTL RR равно 2 дням.

Длина записи ресурса (resource data length) указывает на количество данных ресурса (resource data). Формат этих данных зависит от типа (type). Для типа равного 1 (запись A) данные ресурса – это 4-байтный IP адрес.

3. Процедура обработки стандартного запроса адреса.

В этом примере указано разборщику на узле sun использовать сервер DNS на узле noao.edu (140.252.1.54). На рисунке 9 показано взаимное расположение этих трех систем.

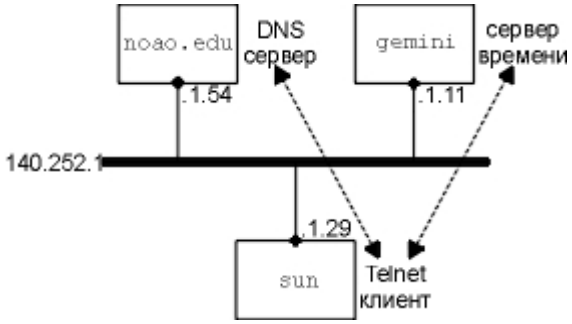

Рисунок 9.

Разборщик является частью клиента. Он устанавливает контакт с сервером DNS, чтобы получить IP адрес, перед тем как будет установлено TCP соединение между Telnet клиентом и сервером времени.

Файл /etc/resolv.conf на узле sun сообщает разборщику о необходимости сделать следующее:

```
sun % cat /etc/resolv.conf
nameserver 140.252.1.54
domain tuc.noao.edu
```

Первая строка сообщает IP адрес DNS сервера – узла noao.edu. Может быть указано до трех строк nameserver, таким образом, будет обеспечен запасной сервер на случай, если один из них выключен или недоступен. Строка domain содержит домен по умолчанию. Если искомое имя не является полным именем домена (не заканчивается точкой), к имени добавляется имя домена по умолчанию .tuc.noao.edu.

На рисунке 10 показан обмен пакетами между разборщиком и сервером DNS.

```
1  0.0                    140.252.1.29.1447 > 140.252.1.54.53: 1+ A?
                              gemini.tuc.noao.edu. (37)
2  0.290820 (0.2908)          140.252.1.54.53 > 140.252.1.29.1447: 1* 2/0/0 A
                         140.252.1.11 (69)
```

Рисунок 10.

Начиная со строки 1, поле после двоеточия (1+) означает, что поле идентификации равно 1, а знак плюс обозначает, что установлен флаг RD (требуется рекурсия). По умолчанию разборщик требует рекурсию.

Следующее поле, A?, означает, что тип запроса – A (необходимо получить IP адрес), а маркировка вопроса обозначает, что это запрос (не ответ). Затем печатается имя запроса: gemini.tuc.noao.edu.. Разборщик добавляет последнюю точку к имени запроса, указывая на то, что это абсолютное имя домена.

Длина пользовательских данных в UDP датаграмме составляет 37 байт: 12 байт – заголовок фиксированного размера (рисунок 3), 21 байт – имя запроса (рисунок 6) и 4 байта – тип запроса и класс запроса. То что UDP датаграмма имеет нечетную длину напоминает нам, что в DNS сообщениях не используются биты заполнения.

Строка 2 в выводе команды tcpdump это ответ от DNS сервера, где 1* в поле идентификации со звездочкой обозначает, что установлен флаг AA (авторитетный ответ). (Так как первичный сервер для домена noao.edu имеет полное представление об именах внутри этого домена.)

Вывод 2/0/0 показывает количество записей ресурсов в трех последних полях с переменной длиной отклика: 2 ответ RR, 0 полномочия RR и 0 дополнительные RR. Команда tcpdump печатает только первый ответ, который в данном случае имеет тип A (IP адрес) со значением 140.252.1.11.

И последняя деталь, на которую необходимо обратить внимание в этом примере, это размер UDP данных в отклике: 69 байт. Чтобы объяснить эту величину, надо знать две вещи.

1.Вопрос возвращается в отклике.

2.При отправке отклика с именами доменов может быть использовано множество повторов. Поэтому используется схема сжатия. И действительно имя домена gemini.tuc.noao.edu появляется трижды. Схема сжатия довольно проста. Везде, где в имени домена появляется метка, используется единственный байт-счетчик (который находится в диапазоне от 0 до 63), у которого два старших бита установлены в 1. Это 16-битный указатель, а не 8-битный байт-счетчик. Следующие 14 байт в указателе определяют смещение следующей метки в DNS сообщении. (Смещение первого байта в поле идентификации равно 0). Этот указатель может появиться там, где появляется метка, а не только там, где появляется полное имя домена, однако возможно, что указатель будет иметь как форму полного имени домена, так и всего лишь окончательной части имени. (Это потому, что окончательные метки в именах заданных доменов часто бывают идентичны.)

На рисунке 11 показан формат DNS отклика, что соответствует строке 2 на рисунке 10. Здесь показаны IP и UDP заголовки, чтобы напомнить о том, что DNS сообщения обычно инкапсулируются в UDP датаграммы. Два возвращенных ответа одинаковы, за исключением различных IP адресов. В этом примере каждый указатель в ответе имеет значение 12, что является смещением от начала DNS заголовка полного имени домена.

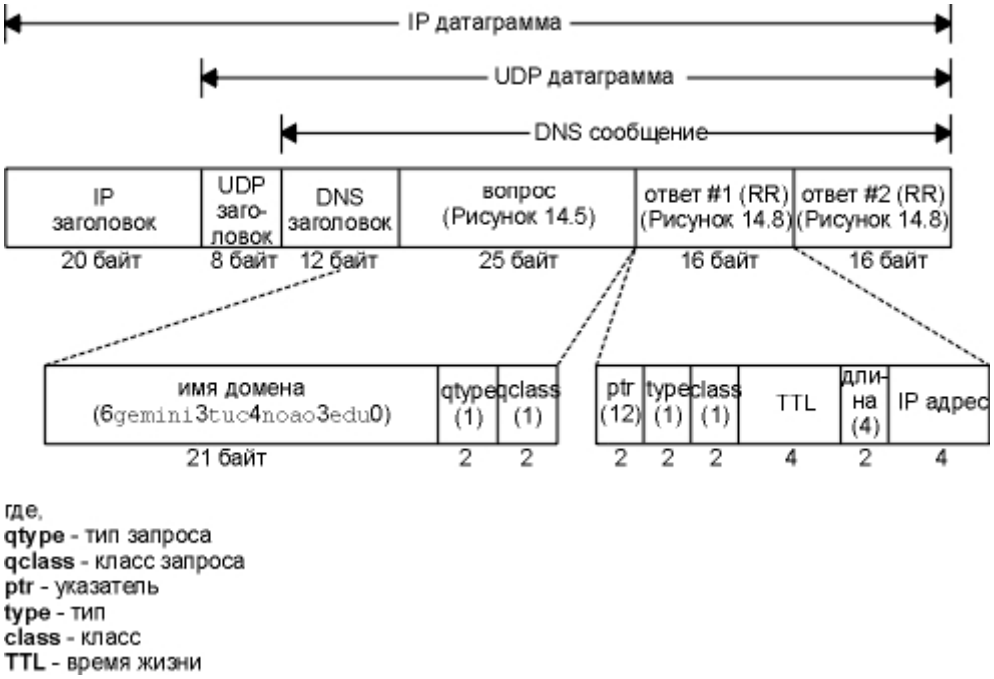

Рисунок 11

4. Инверсные запросы DNS.

Для понимания работы DNS важно знать, как обрабатываются запросы указателя – задан IP адрес, возвращается имя (или имена), соответствующее этому адресу.

Во-первых, вернемся к рисунку 1 и рассмотрим домен верхнего уровня arpa, а также домен in-addr, находящийся ниже. Когда организация вступает в Internet и получает часть пространства имен DNS, она также получает право на часть пространства имен in-addr.arpa, соответствующее ее IP адресам в Internet. В данном случае noao.edu – это сеть класса B с идентификатором 140.252. Уровень дерева DNS ниже in-addr.arpa должен быть первым байтом IP адреса (140 в данном примере), следующий уровень это следующий байт IP адреса (252), и так далее. Однако помните, что имена пишутся, снизу-вверх по дереву DNS. Это означает, что DNS имя узла sun с IP адресом 140.252.13.33 будет 33.13.252.140.in-addr.arpa.

Необходимо писать 4 байта IP адреса задом наперед, потому что полномочия делегируются на основе идентификаторов сетей: первый байт адрес класса A, первый и второй байты адреса класса B, а первый, второй и третий байты это адреса класса C. Первый байт IP адреса должен быть непосредственно под меткой in-addr, однако полные имена доменов (FQDN) пишутся снизу вверх по дереву. Если бы FQDN писались сверху вниз, DNS имя для IP адреса было бы arpa.in-addr.140.252.13.33, однако в этом случае FQDN для узла должно быть edu.noao.tuc.sun.

Без отдельных ветвей дерева DNS осуществить преобразования адрес – имя, (обратное преобразование) можно было бы только начиная от корня дерева и просматривая каждый домен верхнего уровня. При сегодняшнем размере Internet это могло бы занять дни или даже недели. Использование же in-addr.arpa приемлемый вариант, несмотря на переставленные местами байты в IP адресе и специальные домены, иногда вносящие определенную путаницу.

5. Типы RR записей.

Всего существует около 20 различных типов записей ресурсов.

A

Запись A определяет IP адрес. Хранится как 32-битное двоичное значение.

PTR

Запись указателя используется для запросов указателя. IP адрес представляется в виде имени домена (последовательность меток) в домене in-addr.arpa.

CNAME

"Каноническое имя" (canonical name). Представляется как имя домена (последовательность меток). Имя домена, которое имеет каноническое имя, часто называется псевдонимом (alias). Они используются некоторыми FTP узлами, для того чтобы предоставить легкозапоминаемый псевдоним для какой-либо системы.

HINFO

Информация о узле: две символьные строки, указывающие на центральный процессор (CPU) и операционную систему. Не все узлы предоставляют записи HINFO для своих систем, и предоставляемая информация может быть устаревшей.

MX
Записи, посвященные обмену почтой.

NS
Запись имени сервера. Указывает на полномочные DNS серверы для домена. Представлена в виде имен доменов (последовательность меток).

6. Кэширование DNS.

Чтобы уменьшить трафик DNS в Internet, все сервера DNS используют кэширование. В стандартных Unix реализациях кэш поддерживается сервером, а не разборщиком. Так как разборщик является частью каждого приложения, а приложения приходят и уходят, оставляя кэш в программах, которые живут все время, пока система работает (сервер DNS), имеет смысл поддерживать кэш именно на сервере. При этом кэш доступен любому приложению, которое использует сервер. Любые другие узлы в узле, которые используют этот сервер DNS, также пользуются кэшем сервера.

Реализация DNS сервера на базе ОС FreeBSD UNIX

По умолчанию во FreeBSD используется одна из версий программы BIND (Berkeley Internet Name Domain), являющейся самой распространенной реализацией протокола DNS.

Во FreeBSD демон BIND, по очевидным причинам, называется **named**.

Файл	Описание
`named(8)`	Демон BIND
`rndc(8)`	Программа управления демоном сервера имён
`/etc/namedb`	Каталог, в котором располагается вся информация о зонах BIND
`/etc/namedb/named.conf`	Конфигурационный файл для демона

Файлы зон обычно располагаются в каталоге `/etc/namedb` и содержат информацию о зоне DNS, за которую отвечает сервер имён.

В зависимости от способа конфигурации зоны на сервере файлы зон могут располагаться в подкаталогах `master`, `slave` или `dynamic` иерархии`/etc/namedb`. Эти файлы содержат DNS информацию, которую и будет сообщать в ответ на запросы сервер имен.

1. Запуск BIND.

Стандартная конфигурация **named** запускает простой кэширующий сервер в ограниченной среде chroot(8). Для одноразового запуска демона в этой конфигурации используйте команду

```
# /etc/rc.d/named forcestart
```

Чтобы демон **named** запускался во время загрузки, поместите в `/etc/rc.conf` следующую строку:

```
named_enable="YES"
```

Разумеется, существует множество различных конфигураций `/etc/namedb/named.conf`. Разнообразные опции запуска **named** во FreeBSD описаны в переменных `named_*` файла `/etc/defaults/rc.conf`.

2. Конфигурационные файлы.

Файлы конфигурации демона **named** расположены в каталоге `/etc/namedb` и, за исключением случая, когда вам требуется просто резолвер, требуют модификации.

3. Использование `make-localhost`.

Для создания основной зоны для локального узла необходимо перейти в каталог `/etc/namedb` и выполнить команду

```
# sh make-localhost
```

В каталоге `master` должны появиться файлы `localhost.rev` для локальной адресной зоны и `localhost-v6.rev` для для конфигурации IPv6. Ссылки на эти файлы уже содержатся в файле конфигурации `named.conf`.

Это примеры описаний прямой и обратной зон из файла `named.conf` для вторичных серверов.

Для каждой новой зоны, которую будет обслуживать сервер имён, в файл `named.conf` должна быть добавлена запись. К примеру, самая простая запись для домена `example.org` может выглядеть вот так:

```
zone "example.org" {
    type master;
    file "master/example.org";
};
```

Зона является первичной, что отражается в поле `type`, и информация о зоне хранится в файле `/etc/namedb/master/example.org`, что указывается в поле `file`.

```
zone "example.org" {
    type slave;
    file "slave/example.org";
};
```

В случае вторичной зоны информация о ней передается с основного сервера имён для заданной зоны и сохраняется в указанном файле. Если и когда основной сервер имён выходит и строя или недосягаем, то скачанная информация о зоне будет находиться на вторичных серверах, и они смогут обслуживать эту зону.

4. Файлы зон.

Пример файла зоны `example.org` для основного сервера (располагающийся в файле `/etc/namedb/master/example.org`) имеет такой вид:

```
$TTL 3600        ; 1 hour
example.org.    IN      SOA      ns1.example.org. admin.example.org. (
                                2006051501      ; Serial
                                10800           ; Refresh
                                3600            ; Retry
                                604800          ; Expire
                                86400           ; Minimum TTL
                        )

; DNS Servers
                IN      NS      ns1.example.org.
                IN      NS      ns2.example.org.

; MX Records
                IN      MX 10   mx.example.org.
                IN      MX 20   mail.example.org.

                IN      A       192.168.1.1

; Machine Names
localhost       IN      A       127.0.0.1
ns1             IN      A       192.168.1.2
ns2             IN      A       192.168.1.3
mx              IN      A       192.168.1.4
mail            IN      A       192.168.1.5

; Aliases
www             IN      CNAME   @
```

Все имена узлов, оканчивающиеся на ".", задают полное имя, тогда как все имена без символа "." на конце считаются заданными относительно origin. Например, `www` преобразуется в `www.origin`. В нашем файле ориджином является `example.org.`, так что `www` преобразуется в `www.example.org.`

Файл зоны имеет следующий формат:

```
recordname      IN recordtype  value
```

Наиболее часто используемые записи DNS:

SOA

начало зоны ответственности

NS

авторитативный сервер имен

A

адрес узла

CNAME

каноническое имя для алиаса

MX

обмен почтой

PTR

указатель на доменное имя (используется в обратных зонах DNS)

```
example.org. IN SOA ns1.example.org. admin.example.org. (
                        2006051501      ; Serial
                        10800           ; Refresh after 3 hours
                        3600            ; Retry after 1 hour
                        604800          ; Expire after 1 week
                        86400 )         ; Minimum TTL of 1 day
```

`example.org.`

имя домена, а также ориджин для этого файла зоны.

`ns1.example.org.`

основной/авторитативный сервер имён для этой зоны.

`admin.example.org.`

человек, отвечающий за эту зону, адрес электронной почты с символом "@" замененным на точку. (`<admin@example.org>` становится`admin.example.org`)

`2006051501`

последовательный номер файла. При каждом изменении файла зоны это число должно увеличиваться. В настоящее время для нумерации многие администраторы предпочитают формат `ггггммддвв`. `2006051501` будет означать, что файл последний раз изменялся 15.05.2006, а последнее число `01` означает, что это была первая модификация файла за день. Последовательный номер важен, так как он служит для того, чтобы вторичные серверы узнавали об обновлении зоны.

```
IN      NS      ns1.example.org.
```

Это NS-запись. Такие записи должны иметься для всех серверов имён, которые будут отвечать за зону.

```
localhost       IN      A       127.0.0.1
ns1             IN      A       192.168.1.2
ns2             IN      A       192.168.1.3
mx              IN      A       192.168.1.4
mail            IN      A       192.168.1.5
```

Записи типа A служат для обозначения имён машин. Как это видно выше, имя `ns1.example.org` будет преобразовано в `192.168.1.2`.

```
IN      A       192.168.1.1
```

Эта строка присваивает IP адрес `192.168.1.1` текущему ориджину, в данном случае домену `example.org`.

```
www          IN CNAME    @
```

Записи с каноническими именами обычно используются для присвоения машинам псевдонимов. В этом примере `www` является псевдонимом для "главной" машины, соответствующей ориджину, то есть `example.org` (`192.168.1.1`). Записи CNAME могут использоваться для присвоения псевдонимов именам узлов или для использования одного имени несколькими машинами по очереди.

```
IN MX   10      mail.example.org.
```

MX-запись указывает, какие почтовые серверы отвечают за обработку входящей электронной почты для зоны. `mail.example.org` является именем почтового сервера, а 10 обозначает приоритет этого почтового сервера.

Можно иметь несколько почтовых серверов с приоритетами, например, 10, 20 и так далее. Почтовый сервер, пытающийся доставить почту для`example.org`, сначала попробует связаться с машиной, имеющий MX-запись с самым

большим приоритетом (наименьшим числовым значением в поле MX), затем с приоритетом поменьше и так далее, до тех пор, пока почта не будет отправлена.

Для файлов зон in-addr.arpa (обратные записи DNS) используется тот же самый формат, отличающийся только использованием записей PTR вместо A или CNAME.

```
$TTL 3600

1.168.192.in-addr.arpa. IN SOA ns1.example.org. admin.example.org. (
                        2006051501      ; Serial
                        10800           ; Refresh
                        3600            ; Retry
                        604800          ; Expire
                        3600 )          ; Minimum

        IN      NS      ns1.example.org.
        IN      NS      ns2.example.org.

1       IN      PTR     example.org.
2       IN      PTR     ns1.example.org.
3       IN      PTR     ns2.example.org.
4       IN      PTR     mx.example.org.
5       IN      PTR     mail.example.org.
```

В этом файле дается полное соответствие имён узлов IP-адресам в нашем описанном ранее вымышленном домене.

Задание на работу

1. Произвести настройку первичного сервера DNS для зоны согласно варианту задания. Определить резервный сервер DNS для созданной зоны. Выполнить процедуру передачи информации о зоне.
2. Использовать сервер в качестве резервного для заданной зоны.

Варианты заданий.

№ варианта	Имя зоны	Резервный сервер	Почтовый шлюз	Узлы		
				Имя	Адрес	Псевдоним
1	zone01.com.ua	10.18.51.2	mail	alpha	172.20.1.10	ws1
				beta	172.20.1.20	ws2
				gamma	172.20.1.30	ws3
				delta	172.20.1.40	ws4
				omega	172.20.1.50	ws5
2	zone02.net.ua	10.18.51.3	smtp	mercury	192.168.11.21	srv-01
				venus	192.168.11.22	srv-02
				earth	192.168.11.23	srv-03
				saturn	192.168.11.24	srv-04
				jupiter	192.168.11.25	srv-05
3	zone03.kiev.ua	10.18.51.1	mail	tiger	192.168.1.11	www
				lion	192.168.1.12	ftp
				lynx	192.168.1.13	nntp
				leopard	192.168.1.14	pop3
				jaguar	192.168.1.15	imap
4	zone04.com	10.18.51.5	smtp	rose	172.20.1.31	machine-1
				tulip	172.20.2.32	machine-2
				narcissus	172.20.3.33	machine-3
				aster	172.20.4.34	machine-4
				peony	172.20.5.35	machine-5
5	zone05.net	10.18.51.6	mail	alpha	172.25.11.10	srv-01
				beta	172.25.11.20	srv-02
				gamma	172.25.11.30	srv-03
				delta	172.25.11.40	srv-04
				omega	172.25.11.50	srv-05
6	zone06.org.ua	10.18.51.4	smtp	mercury	192.168.22.10	www
				venus	192.168.22.20	ftp
				earth	192.168.22.30	nntp

				saturn	192.168.22.40	pop3
				jupiter	192.168.22.50	imap
7	zone07.org	10.18.51.8	mail	tiger	172.30.10.31	machine-1
				lion	172.30.10.32	machine-2
				lynx	172.30.10.33	machine-3
				leopard	172.30.10.34	machine-4
				jaguar	172.30.10.35	machine-5
8	zone08.edu	10.18.51.9	smtp	rose	192.168.33.1	ws1
				tulip	192.168.33.2	ws2
				narcissus	192.168.33.3	ws3
				aster	192.168.33.4	ws4
				peony	192.168.33.5	ws5
9	zone09.org	10.18.51.7	mail	alpha	192.168.55.10	www
				beta	192.168.55.20	ftp
				gamma	192.168.55.30	nntp
				delta	192.168.55.40	pop3
				omega	192.168.55.50	imap
10	zone10.org.ua	10.18.51.11	smtp	mercury	172.21.11.10	machine-1
				venus	172.21.11.20	machine-2
				earth	172.21.11.30	machine-3
				saturn	172.21.11.40	machine-4
				jupiter	172.21.11.50	machine-5
11	zone11.net	10.18.51.12	mail	tiger	172.31.50.1	ws1
				lion	172.31.50.2	ws2
				lynx	172.31.50.3	ws3
				leopard	172.31.50.4	ws4
				jaguar	172.31.50.5	ws5
12	zone12.com	10.18.51.10	smtp	rose	192.168.77.11	srv-01
				tulip	192.168.77.22	srv-02
				narcissus	192.168.77.33	srv-03
				aster	192.168.77.44	srv-04
				peony	192.168.77.55	srv-05
13	zone13.kiev.ua	10.18.51.14	mail	alpha	192.168.99.10	machine-1
				beta	192.168.99.20	machine-2
				gamma	192.168.99.30	machine-3
				delta	192.168.99.40	machine-4
				omega	192.168.99.50	machine-5
14	zone14.net.ua	10.18.51.15	smtp	mercury	172.16.70.1	ws1
				venus	172.16.70.2	ws2
				earth	172.16.70.3	ws3
				saturn	172.16.70.4	ws4
				jupiter	172.16.70.5	ws5
15	zone15.com.ua	10.18.51.13	mail	tiger	192.168.88.11	srv-01
				lion	192.168.88.12	srv-02
				lynx	192.168.88.13	srv-03
				leopard	192.168.88.14	srv-04
				jaguar	192.168.88.15	srv-05

Контрольные вопросы

Литература

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 3-е изд. – СПб.: Питер, 2007. – 960 с.: ил.

2. http://www.freebsd.org/doc/ru_RU.KOI8-R/books/handbook/network-dns.html

3. Стивенс У.Р. Протоколы TCP/IP. Практическое руководство/ Пер. с англ. и коммент. А.Ю. Глебовского. – СПб.: «Невский диалект» - «БХВ-Петербург», 2003. – 672 с.: ил.