

DNS – повторительный курс



Обзор

- Цели этого курса
- Что такое DNS ?
- DNS: как он сделан и как он работает?
- Как работают запросы?
- Типы записей
- Кэширующие и авторитетные серверы
- Делегирование: домены и зоны
- Поиск ошибок: где находится проблема?

Цели этого курса

- Освежить основы DNS, включая механизмы запросов, делегирование, и кэширование.
- Добиться понимания DNS, достаточного для конфигурации кэширующего DNS сервера, и для отладки обычных проблем DNS, как локальных, так и удаленных (в сети Internet)

Что такое DNS ?

- Система для преобразования имен в IP адреса:

nsrc.org → 128.223.157.19
www.afrinic.net → 2001:42d0::200:80:1

- ... и обратно:

128.223.157.19 → nsrc.org
1.0.0.0.0.8.0.0.0.0.2.0.0.0.0.0.0.0.0.0
.0.0.0.0.0.d.2.4.1.0.0.2.ip6.arpa. →
www.afrinic.net.

Что такое DNS ?

- DNS предоставляет и другую информацию:
 - куда направлять почту для домена
 - кто отвечает за какую-либо систему
 - географическая информация
 - и т.д. . . .
- Как находить всю эту информацию?

Основные инструменты DNS

- Использование команды host:

```
# host nsrc.org.
```

```
nsrc.org. has address 128.223.157.19
```

```
# host 128.223.157.19
```

```
19.157.223.128.in-addr.arpa domain name  
pointer nsrc.org.
```

Основные инструменты DNS

- Сервер с IPv6:

```
# host www.afrinic.net
```

www.afrinic.net has IPv6 address

2001:42d0::200:80:1

```
# host 2001:42d0::200:80:1
```

```
1.0.0.0.8.0.0.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.  
0.0.0.0.d.2.4.1.0.0.2.ip6.arpa domain name  
pointer www.afrinic.net.
```

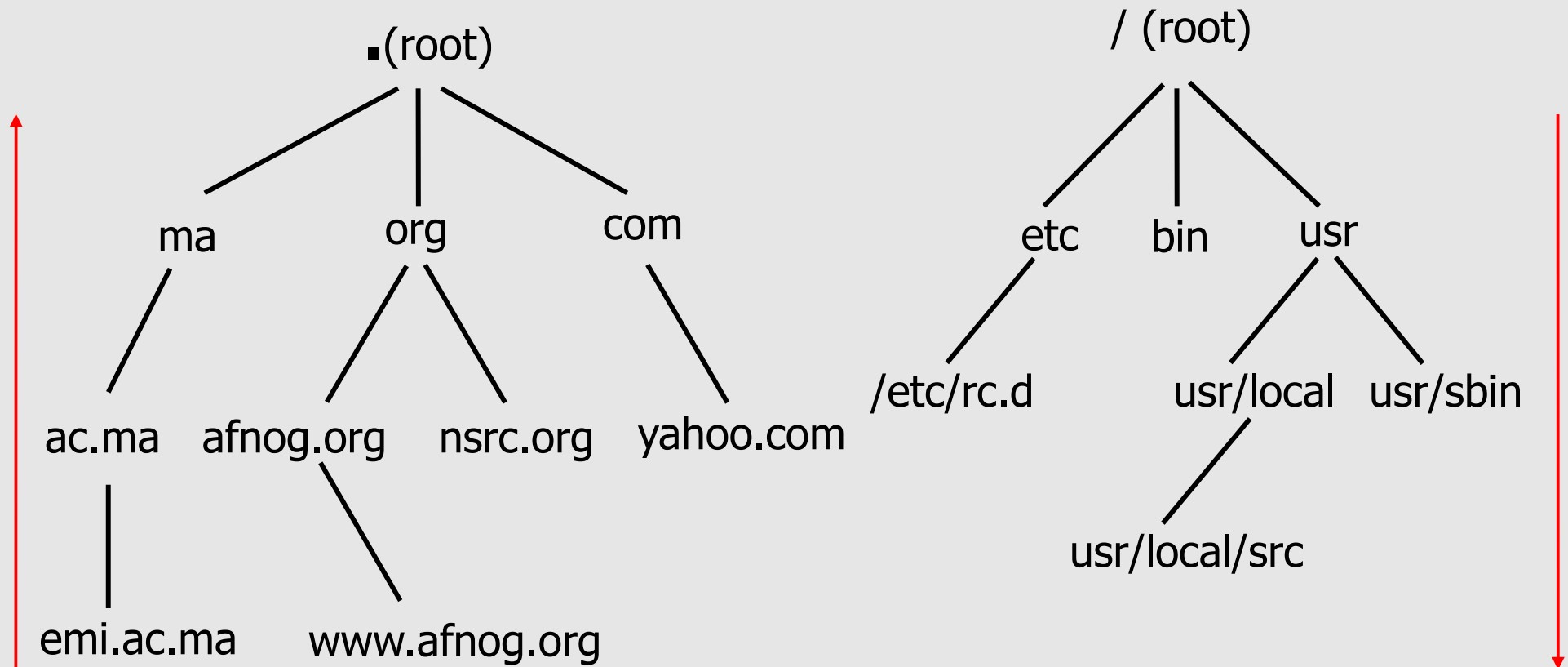
Основные инструменты DNS

- Попробуйте сами с другими именами – вначале найдите нижеследующие имена, потом сделайте то же самое с полученными IP адресами:

`www.yahoo.com`
`www.nsrc.org`
`ipv6.google.com`

- Всегда ли поиск IP соответствует имени? Почему?
- Где команда 'host' находит информацию?

Как устроен DNS?



База данных DNS

Файловая система Unix

... образует дерево

Как устроен DNS?

- DNS имеет иерархическую структуру
- DNS администрируется распределенно — не существует определенного центра, который бы администратировал все данные в DNS
- Такое распределение администрирования называется *делегированием*

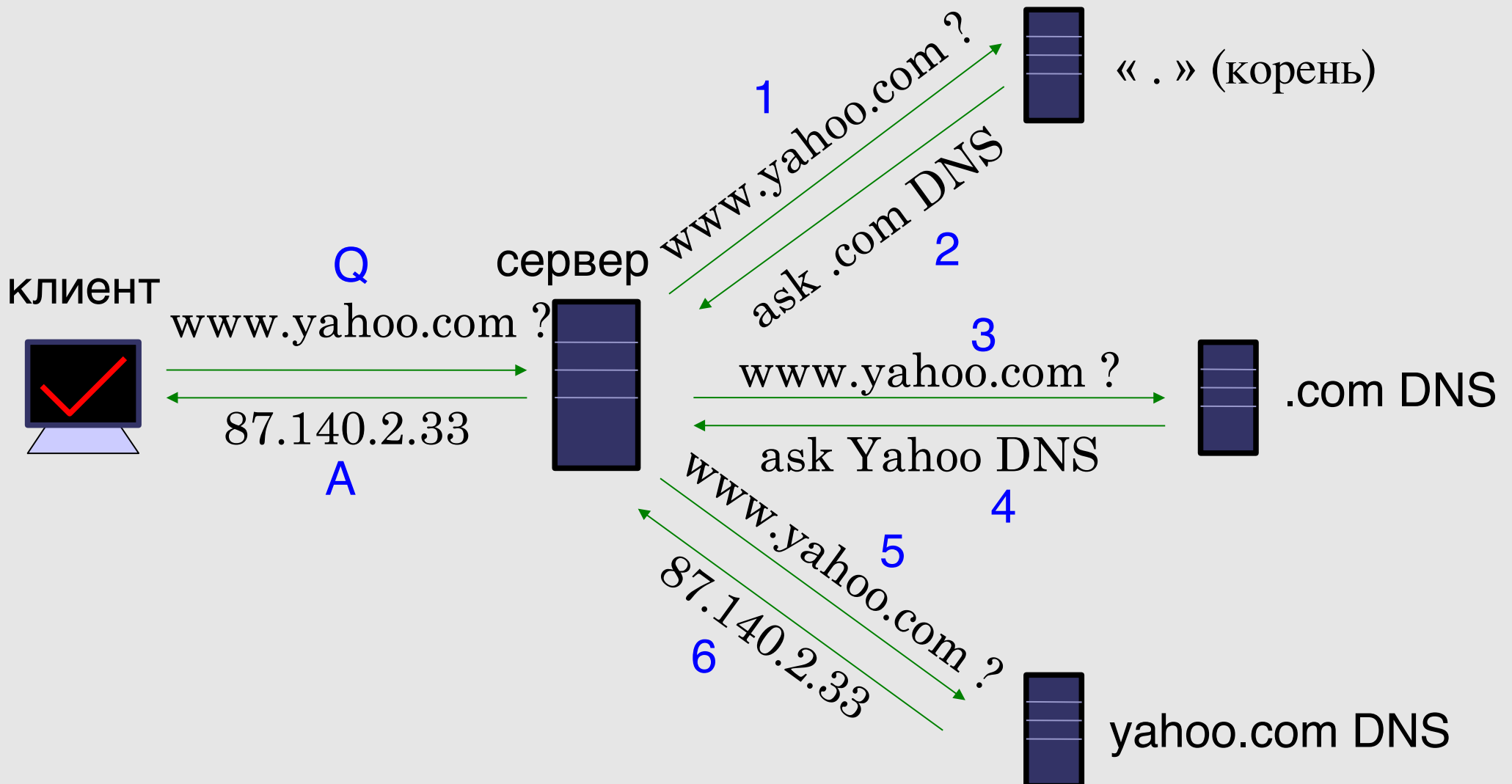
Как работает DNS?

- **Клиенты**, используя механизм, называемый **разрешением имен**, опрашивают **серверы** — это называется **запросом**
- Запрашиваемый сервер попытается найти ответ от имени клиента
- Сервер работает рекурсивно, сверху (от корня) донизу, до тех пока не найдет ответ, опрашивая другие серверы когда нужно — сервер ссылается на другие серверы

Как работает DNS?

- Клиент (web браузер, почтовая программа, ...) использует систему разрешения имен операционной системы для нахождения IP адреса.
- Например, если мы идем на страницу `www.yahoo.com`:
 - Браузер спрашивает OS «Мне нужен IP для `www.yahoo.com`»
 - OS находит из конфигурации системы разрешения имен, какой сервер спросить, и посылает тому запрос
- В UNIX, `/etc/resolv.conf` хранит конфигурацию системы разрешения имен

DNS запрос



Tcpdump деталей запроса

- Станьте администратором на сервере:

```
$ sudo -s
```

```
passwd:
```

```
# tcpdump -s1500 -n port 53
```

- В другом окне или экране:

```
# host ... (что угодно)
```

Детали запроса – пример

- 1: 18:40:38.62 IP 192.168.1.1.57811 > 192.112.36.4.53:29030 [1au] A? h1-web.hosting.catpipe.net. (55)
- 2: 18:40:39.24 IP 192.112.36.4.53 > 192.168.1.1.57811:29030- 0/13/16 (540)
- 3: 18:40:39.24 IP 192.168.1.1.57811 > 192.43.172.30.53:7286 [1au] A? h1-web.hosting.catpipe.net. (55)
- 4: 18:40:39.93 IP 192.43.172.30.53 > 192.168.1.1.57811:7286 FormErr- [0q] 0/0/0 (12)
- 5: 18:40:39.93 IP 192.168.1.1.57811 > 192.43.172.30.53:50994 A? h1-web.hosting.catpipe.net. (44)
- 6: 18:40:40.60 IP 192.43.172.30.53 > 192.168.1.1.57811:50994- 0/3/3 (152)
- 7: 18:40:40.60 IP 192.168.1.1.57811 > 83.221.131.7.53:58265 [1au] A? h1-web.hosting.catpipe.net. (55)
- 8: 18:40:41.26 IP 83.221.131.7.53 > 192.168.1.1.57811:58265* 1/2/3 A 83.221.131.6 (139)

Детали запроса - анализ

- Мы используем анализатор пакетов (wireshark) для того, чтобы просмотреть содержимое запроса...

<http://www.wireshark.org/>

The screenshot displays the Wireshark interface with a list of captured packets. The selected packet (No. 10) is an HTTP retransmission. The detailed view shows the packet structure: Ethernet II, Internet Protocol (IPv4), Transmission Control Protocol (TCP), and Hypertext Transfer Protocol (HTTP).

No.	Time	Source	Destination	Protocol	Info
1	0.000000	69.4.231.52	10.10.2.171	HTTP	Continuation or non-HTTP traffic
2	0.000477	10.10.2.171	69.4.231.52	TCP	43076 > http [ACK] Seq=1 Ack=429
3	0.026605	Olicom_cb:4f	Broadcast	ARP	Who has 10.10.2.168? Tell 10.10
4	0.073463	Netgear_97:7f	Spanning-tree-(f	STP	Conf. Root = 32768/0/00:0f:b5:97
5	0.074800	Olicom_cb:4f	Broadcast	ARP	Who has 10.10.2.168? Tell 10.10
6	0.206011	Olicom_cb:4f	Broadcast	ARP	Who has 10.10.2.168? Tell 10.10
7	0.207065	10.10.2.178	10.10.2.255	NBNS	Name query NB WVLKA0<1c>
8	0.214690	fe80::8d4a:d	ff02::1:2	DHCPv6	Solicit
9	0.224232	10.10.2.180	239.255.255.250	SSDP	M-SEARCH * HTTP/1.1
10	0.290652	69.4.231.52	10.10.2.171	HTTP	[TCP Retransmission] Continuation
11	0.291095	10.10.2.171	69.4.231.52	TCP	43076 > http [ACK] Seq=1 Ack=144
12	0.444050	10.10.2.166	10.10.2.171	DNS	Standard query & tertiary dnc

Frame 1 (1514 bytes on wire, 1500 bytes captured)

- Ethernet II, Src: Olicom_cb:4f:a2 (00:00:24:cb:4f:a2), Dst: HewlettP_8c:91:8b (00:1a:4b:8c:91:8b)
- Internet Protocol, Src: 69.4.231.52 (69.4.231.52), Dst: 10.10.2.171 (10.10.2.171)**
- Transmission Control Protocol, Src Port: http (80), Dst Port: 43076 (43076), Seq: 1, Ack: 1, Len: 1448
- Hypertext Transfer Protocol
- [Packet size limited during capture: HTTP truncated]

Hex dump and ASCII view of the packet data:

```
0000 00 1a 4b 8c 91 8b 00 00 24 cb 4f a2 08 00 45 00 ..K....$.0...E.
0010 05 dc 78 cb 40 00 2b 06 00 00 45 04 e7 34 0a 0a ...x.@.+...E..4..
0020 02 ab 00 50 a8 44 d7 a2 c9 69 10 82 fb b9 80 10 ...P.D...i.....
0030 00 0e 45 ff 00 00 01 01 08 0a 1c b5 73 73 00 06 ..E.....ss...
0040 c2 39 86 c9 d5 24 88 cd 1e 3b 5e 1f 97 e8 e6 fd .9...$....;^....
0050 d3 7a 51 bd 8f 53 2a d9 dd c1 8f 13 27 6d 93 74 .70..S*....'m.t
```

Конфигурация разрешения имен

- Откуда ваш компьютер знает, какой сервер запроить для получения ответов на DNS запросы?
- В UNIX, смотрите `/etc/resolv.conf`
- Сейчас загляните в этот файл, и убедитесь, что в нем присутствует строка типа:

```
nameserver a.b.c.d
```

или

```
nameserver ip:v6:ad:dr:es:ss
```

... где `a.b.c.d` - IP/IPv6 адрес рабочего сервера DNS.

Нахождение корня...

- Первый запрос адресован к:

192.112.36.4 (G.ROOT-SERVERS.NET.)
- Откуда серверу известно, по каким адресам расположены корневые серверы?
- Проблема курицы и яйца
- Каждый DNS сервер хранит список корневых серверов и их IP адресов
- В сервере BIND, `named.root`

Более детальная картина при помощи 'dig'

- команда 'host' несколько ограничена — она хороша для поиска, но недостаточна для отладки.
- команда 'dig' дает больше деталей
- dig показывает много интересных вещей...

Более детальная картина при помощи 'dig'

```
ns# dig @147.28.0.39 www.nsrc.org. a
```

```
; <<>> DiG 9.3.2 <<>> @147.28.0.39 www.nsrc.org
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4620
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 4,
ADDITIONAL: 2

;; QUESTION SECTION:
;www.nsrc.org.                IN      A

;; ANSWER SECTION:
www.nsrc.org.                 14400   IN      A      128.223.162.29

;; AUTHORITY SECTION:
nsrc.org.                     14400   IN      NS      rip.psg.com.
nsrc.org.                     14400   IN      NS      arizona.edu.

;; ADDITIONAL SECTION:
rip.psg.com.                  77044   IN      A      147.28.0.39
arizona.edu.                  2301    IN      A      128.196.128.233

;; Query time: 708 msec
;; SERVER: 147.28.0.39#53(147.28.0.39)
;; WHEN: Wed May 10 15:05:55 2007
;; MSG SIZE rcvd: 128
```

```
noc# dig www.afrinic.net any
```

```
; <<>> DiG 9.4.2 <<>> any www.afrinic.net
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36019
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 6, ADDITIONAL: 10
```

```
;; QUESTION SECTION:
```

```
;www.afrinic.net.          IN      ANY
```

```
;; ANSWER SECTION:
```

```
www.afrinic.net. 477      IN      AAAA    2001:42d0::200:80:1
www.afrinic.net. 65423    IN      A       196.216.2.1
```

```
;; AUTHORITY SECTION:
```

```
afrinic.net.      65324    IN      NS       sec1.apnic.net.
afrinic.net.      65324    IN      NS       sec3.apnic.net.
afrinic.net.      65324    IN      NS       ns1.afrinic.net.
afrinic.net.      65324    IN      NS       tinnie.arin.net.
afrinic.net.      65324    IN      NS       ns.lacnic.net.
afrinic.net.      65324    IN      NS       ns-sec.ripe.net.
```

```
;; ADDITIONAL SECTION:
```

```
ns.lacnic.net.    151715    IN      A       200.160.0.7
ns.lacnic.net.    65315    IN      AAAA    2001:12ff::7
ns-sec.ripe.net.  136865    IN      A       193.0.0.196
ns-sec.ripe.net.  136865    IN      AAAA    2001:610:240:0:53::4
ns1.afrinic.net.  65315    IN      A       196.216.2.1
tinnie.arin.net.  151715    IN      A       168.143.101.18
sec1.apnic.net.   151715    IN      A       202.12.29.59
sec1.apnic.net.   151715    IN      AAAA    2001:dc0:2001:a:4608::59
sec3.apnic.net.   151715    IN      A       202.12.28.140
sec3.apnic.net.   151715    IN      AAAA    2001:dc0:1:0:4777::140
```

```
;; Query time: 1 msec
```

```
;; SERVER: 196.200.218.1#53(196.200.218.1)
```

```
;; WHEN: Tue May 27 08:48:13 2008
```

```
;; MSG SIZE rcvd: 423
```

Вывод команды dig

- Интересные поля:
 - раздел флагов: qr aa ra rd
 - статус
 - раздел ответа
 - авторитетный раздел
 - TTL (числа в левой колонке)
 - время выполнения запроса
 - сервер
- Обратите внимание на типы записей 'A' и 'AAAA' в выводе.

Типы записей

- Основные типы записей:
- A, AAAA: IPv4, IPv6 адреса
- NS: DNS сервер
- MX: почтовый сервер
- CNAME: каноническое имя (псевдоним)
- PTR: Реверсивная информация

Кэширование и авторитетность

- В выводе команды `dig`, и в последующих выводах, мы заметили уменьшение времени запроса при повторении запроса.
- Ответы **кэшируются** DNS сервером, выполняющим запрос, для ускорения запросов и для экономии ресурсов сети
- Значение TTL управляет временем кэширования
- DNS серверы можно разделить на две категории: **кэширующие** и **авторитетные**.

Авторитетные серверы

- Обычно, авторитетные серверы отвечают только на запросы об информации, для которой они являются авторитетным источником, т.е. об информации, хранящейся на самом сервере (в файле или в базе данных)
- Если они не знают ответа, они укажут на авторитетный источник, но не будут обрабатывать запрос рекурсивно.

Кэширующие серверы

- Кэширующие серверы перенаправляют запросы от имени клиента, и кэшируют ответы на будущее.
- Кэширование и авторитетность могут быть реализованы в одной и той же программе, но смешивание функциональности не рекомендуется (риски защиты данных + путаница)
- TTL ответа определяет, как долго ответ будет кэширован без повторения рекурсивного запроса.

Значения TTL

- Уменьшение значения TTL и истечение срока кэширования
- Сделайте несколько раз запрос об записи типа A для `www.yahoo.com`:

```
# dig www.yahoo.com
```

- Что вы наблюдаете относительно времени запроса и значения TTL?

SOA

- Давайте спросим SOA для домена:

```
# dig SOA <domain>
...
;; AUTHORITY SECTION:
<domain>. 860 IN SOA ns.<domain>. root.<domain>.
                        200702270 ; serial
                        28800      ; refresh
                        14400      ; retry
                        3600000    ; expire
                        86400      ; neg ttl
...
```

SOA

- Выделены первые два поля:
 - SOA (начало авторитетности), которое администратор установил в имя «исходного» сервера для данных домена (это не всегда соблюдается)
 - RP (ответственное лицо) - email адрес (в котором первое '@' изменено на '.') для контакта в случае технических проблем.

SOA

- Другие поля:
 - `serial`: серийный номер зоны, используется для репликации зоны между двумя серверами DNS
 - `refresh`: как часто реплицирующий сервер должен сверяться с мастер-сервером, чтобы узнать, не изменилась ли зона
 - `retry`: как часто пытаться повторять сверку, если мастер-сервер не отвечает.
 - `expire`: если мастер-сервер не отвечает слишком долго, перестать отвечать на клиентские запросы о зоне.
- Почему `expire` необходим?

Эксплуатация кэширующего сервера

- Иметь работающий кэширующий сервер на локальной машине может быть полезно
- Легко настроить, например на FreeBSD:
 - добавьте `named_enable="YES"` в `/etc/rc.conf`
 - запустите `named`:
`/etc/rc.d/named start`
- Что было бы хорошей проверкой того, что `named` работает?

Эксплуатация кэширующего сервера

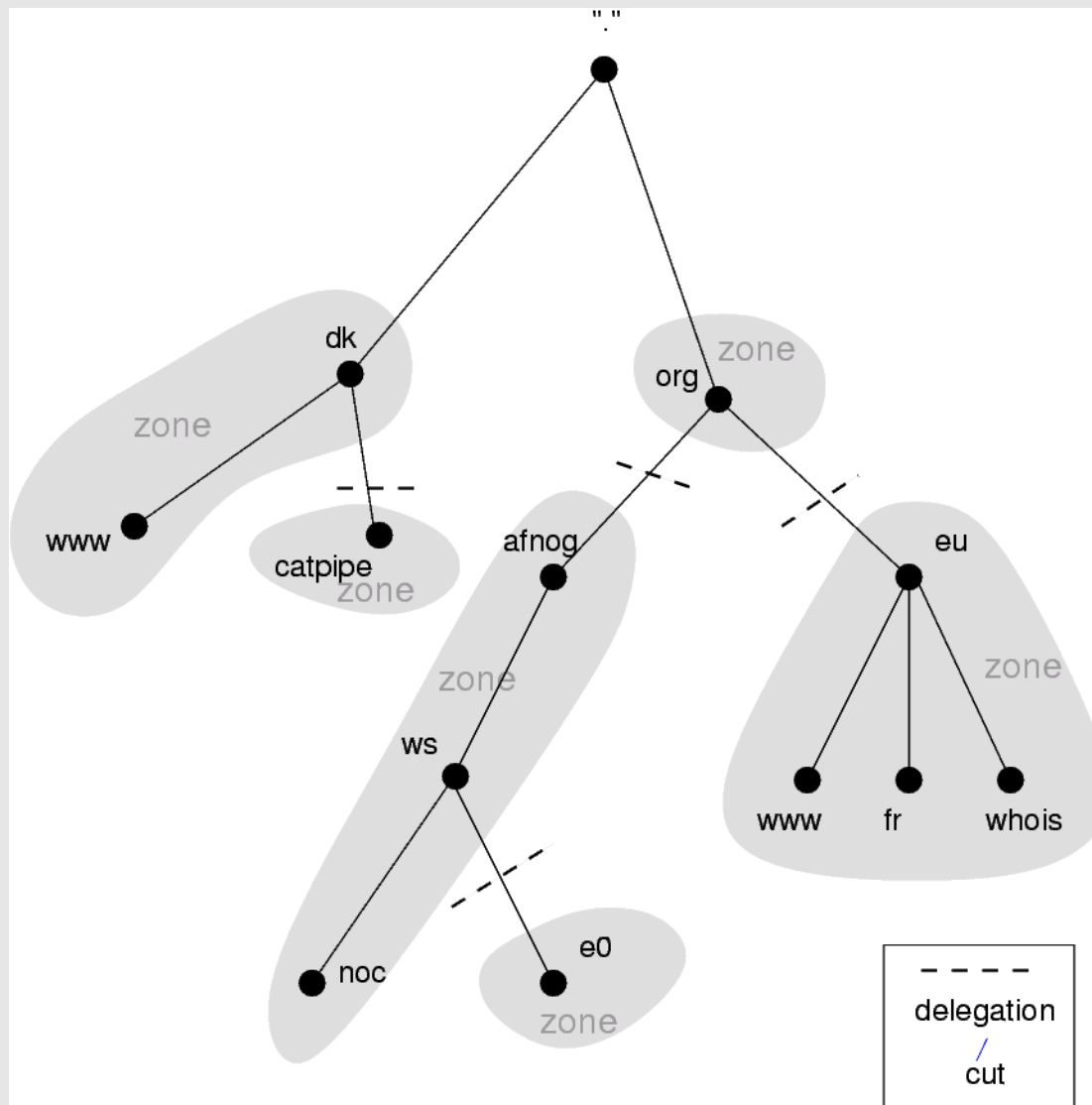
- Когда вы уверены в том, что ваш кэширующий сервер работоспособен, добавьте его в конфигурацию локальной системы разрешения имен (/etc/resolv.conf):

```
nameserver 127.0.0.1
```

Делегирование

- Мы упомянули, что одним из преимуществ DNS является тот факт, что система администрируется распределенно. Это называется делегированием.
- Мы делегируем на границе администрирования, когда мы хотим отдать управление доменом нижнего уровня:
 - отделу внутри большой организации
 - организации внутри страны
 - субъекту, представляющему домен целой страны

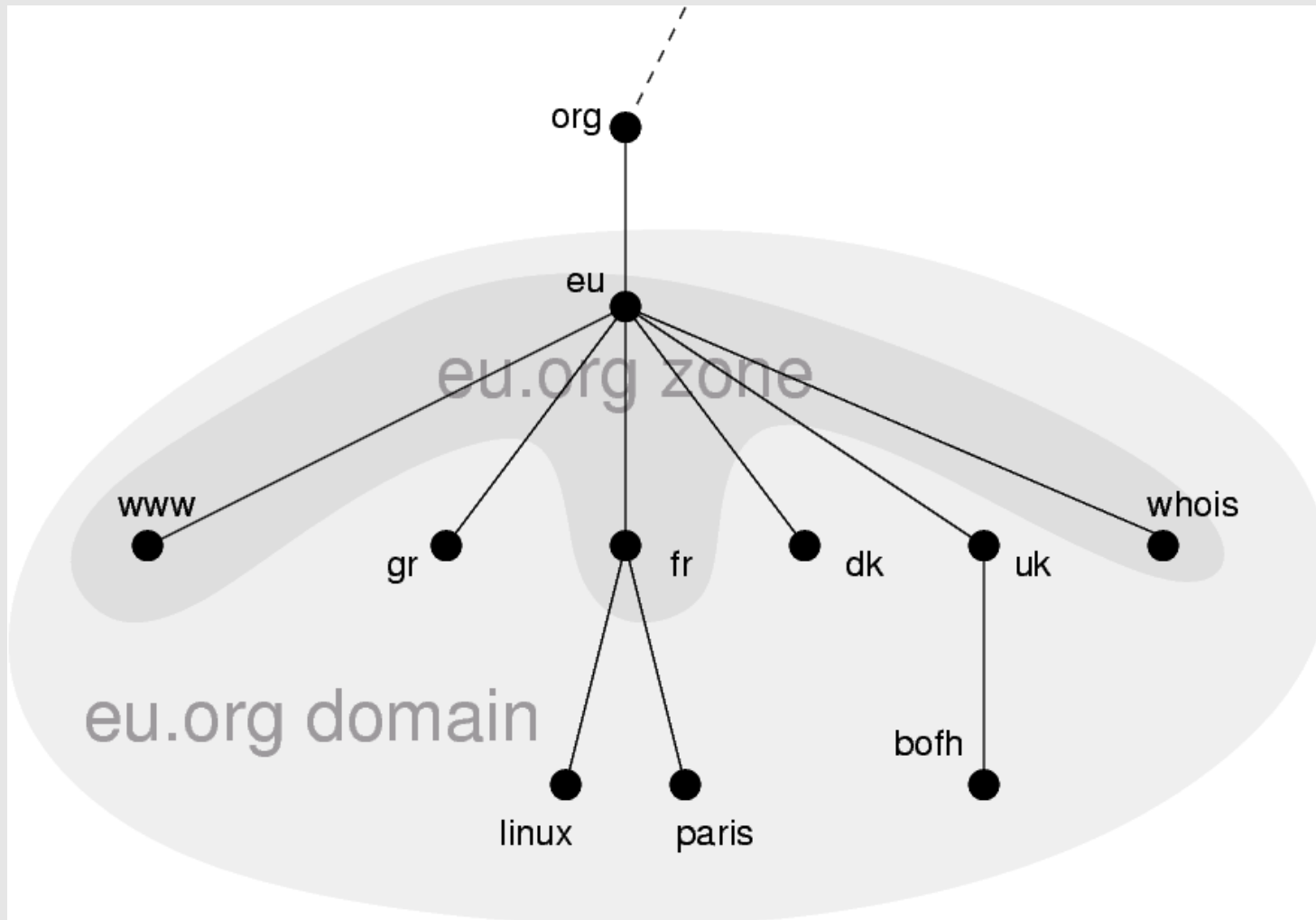
Делегирование



Делегирование: домены и зоны

- Когда мы говорим о целом поддереве, мы говорим о *доменах*
- Когда мы говорим о части домена, администрируемой отдельно, мы говорим о *зонах*

Делегирование: домены и зоны



Поиск ошибок: doc

- Когда вы сталкиваетесь с проблемами с вашей сетью, web сервисом или emailом, вы не всегда подозреваете проблему с DNS.
- Даже если вы думаете, что проблема с DNS, не всегда очевидно, где именно проблема, ибо DNS довольно сложен.
- 'doc' – хороший инструмент для быстрого поиска ошибок конфигурации
- /usr/ports/dns/doc – установите его!
- Давайте с ним немного поиграемся...

Заключение

- DNS – предмет обширный
- Требуется большой опыт для правильной идентификации проблем – кэширование и рекурсия особенно сбивают с толку
- Не забудьте, что существует несколько сервер для одних и тех же данных, и вы не всегда запрашиваете какой-то один из них
- Практика, практика, практика!
- Не бойтесь задавать вопросы...

Вопросы?

?