Domain Name System

William Tevie
tevie@ghana.com
network computer systems

Some DNS topics

- What the Internet's DNS is
- Configuring a resolver on a Unix-like system
- Configuring a nameserver on a Unix-like syst
- Exercises: Create and install a simple zone

What the Internet's DNS is

- A systematic namespace called the domain name s
- Different people or organisations are responsible
- Information is associated with each name
- A set of conventions for using the information
- A distributed database system
- Protocols that allow retrieval of information, and s

A systematic namespace - the domain name space

- Several components (called labels)
  - written separated by dots
  - often written terminated by a dot
- Hierarchical structure
  - Leftmost label has most local scope
  - Rightmost label has global scope
  - Terminal dot represents root of the hierarchy
- Domain names are case independent

Why use hierarchical names?

- Internet hosts and other resources need globally unique names
- Difficult to keep unstructured names unique
  - would require a single list of all names in use
- Hierarchical names are much easier to make unique
  - cat.abc.gh. is different from cat.abc.tg.

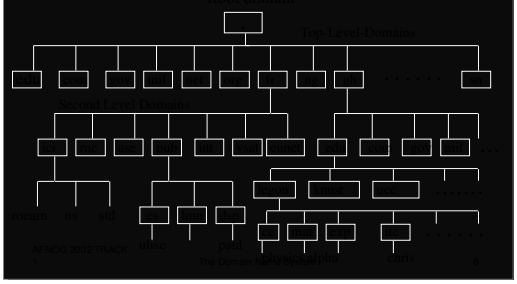
What are domain names used for?

- To identify computers (hosts) on the Internet
  - austin.ghana.com
- To identify organisations
  - afnog.org
- To map other information to a form that is usable with the DNS infrastructure
  - IP addresses, Telephone numbers, AS numbers

Examples of domain names

- .
- COM.
- GH.
- CO.ZA.
- www.afnog.org.
- in-addr.arpa.

Domain Name Hierarchy



Different uses of the term "domain"

- Sometimes, the term "domain" is used to refer to a single name
  - such as www.afnog.org
- Sometimes, the term "domain" is used to refer to all the names (subdomains) that are hierarchically below a particular name
  - in this usage, the afnog.org domain includes www.afnog.org, t1.ws.afnog.org,etc.

Other information mapped to domain names

- Almost any systematic namespace could be mapped to the domain name space
- Need an algorithm agreed to by all people who will use the mapping

Different people responsible for diff. parts

- Administrator responsible for a domain may delegate authority for a subdomain
- Each part that is administered independently is called a zone
- Domain or zone administrator may choose to put subdomains in same zone as parent domain, or in different zone, depending on policy and convenience

The DNS is a distributed database system

- What makes it a distributed database?
- How is data partitioned amongst the servers
- What about reliability?

What makes it a distributed database?

- Thousands of servers around the world
- Each server has authoritative information about some subset of the namespace
- There is no central server that has information about the whole namespace
- If a question gets sent to a server that does not know the answer, that is not a problem

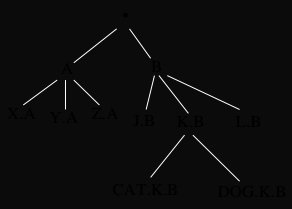
What about reliability?

- If one server does not reply, clients will ask another server
- That's why there are several servers for each zone
- Zone administrators should choose servers that are not all subject to a single point of failure

What is a zone? (1)

- Think of the namespace as a tree or graph of nodes joined by arcs
  - Each node represents a domain name

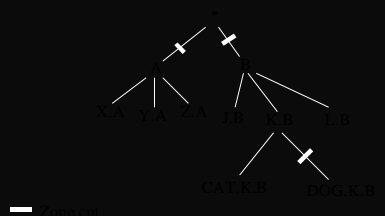
What is a zone? (diagram 1)



What is a zone? (2)

- Think of the namespace as a tree or graph of nodes joined by arcs
  - Each node represents a domain name
- Now cut some of the arcs
  - Each cut represents a delegation of administrative control

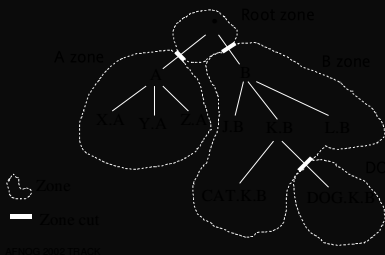
What is a zone? (diagram 2)



What is a zone? (3)

- Each zone consists of a set of nodes that are still joined to each other through paths that do not involve arcs that have been cut
  - The name "CAT.K.B" is in the "B" zone
  - The name "DOG.K.B" is in the "DOG.K.B" zone
  - The "DOG.K.B" zone is a child of the "B" zone

What is a zone? (diagram 3)



Information is associated with each domain name

- Several types of records (Resource Records, RRs), all with a similar format
- Each RR contains some information that is associated with a specific domain name
- Each domain name can have several RRs of the same type or of different types

A set of conventions for using the information

- How to represent the relationship between host names and IP addresses
- What records are used to control mail routing, and how the mail system should use those records
- How to use the DNS to store IP netmask information
- Many other things

General format of RRs

- Owner name - the domain name that this record belongs to
- TTL - how long copies of this RR may be cached (measured in seconds)
- Class - almost always IN
- Type - there are many types
- Data - different RR types have different data formats

Several types of RRs

- IP address for a host
- Information needed by the DNS infrastructure itse
- Hostname for an IP address
- Information about mail routing
- Free form text
- Alias to canonical name mapping
- Many more (but less commonly used)

IP address for a host

- A record
- Owner is host name
- Data is IP address

```
; IP address of austin.gh.com
austin.ghana.com.  86400  IN  A  196.3.64.1
```


Information needed by the DNS infrastructure itself

- SOA record
  - Each zone has exactly one SOA record
- NS records
  - Each zone has several nameservers that are listed as having authoritative information about domains in the zone
  - One NS record for each such nameserver
- Zone cuts are marked by these RRs

SOA record

- Every zone has exactly one SOA record
- The domain name at the top of the zone owns the SOA record
- Data portion of SOA record contains:
  - MNAME – name of master nameserver
  - RNAME – email address of zone administrator
  - SERIAL – serial number
  - REFRESH RETRY EXPIRE MINIMUM – timing parameters

NS record

- Each zone has several listed nameservers
- One NS record for each listed nameserver
  - master/primary and slaves/secondaries
- the data portion of each NS record contains the domain name of a nameserver
- Does not contain IP address
  - Get that from an A record for the nameserver

SOA and NS record example

```
; owner    TTL    class  type   data
ghana.com.  86400  IN     SOA    austin.gh.com. support.gh.com. (
                                  199710161 ; serial
                                  10800     ; refresh after 3 hours
                                  3600      ; retry after 1 hour
                                  604800    ; expire after 1 week
                                  86400     ) ; negative TTL rfc2308
ghana.com.  86400  IN     NS     ns1.ghana.com.
ghana.com.  86400  IN     NS     ns2.ghana.com.
ghana.com.  86400  IN     NS     server.elsewhere.example.
```


SOA and NS example using some shortcuts

```
$ORIGIN ghana.com.
$TTL 86400
; owner    TTL    class  type   data
@                  IN     SOA    austin.gh.com. Support.gh.com. (
                                  199710161 ; serial
                                  10800     ; refresh after 3 hours
                                  3600      ; retry after 1 hour
                                  604800    ; expire after 1 week
                                  86400     ) ; negative TTL rfc2308
                   NS     ns1
                   NS     ns2
                   NS     server.elsewhere.example.
```


Hostname for an IP address

- PTR record
- Owner is IP address, mapped into the in-addr.arpa domain
- Data is name of host with that IP address

```
; host name for IP address 196.3.64.1
1.64.3.196.in-addr.arpa.  PTR  austin.ghana.com.
```


Information about mail routing

- MX record
- Owner is name of email domain
- Data contains preference value, and name of host that receives incoming email

```
; send ghana.com's email to mailserver or backupserver
ghana.com.    MX     0   mail.ghana.com.
ghana.com.    MX     10  backupmail.ghana.com.
```

Alias to canonical name mapping

- CNAME record
- Owner is non-canonical domain name (alias)
- Data is canonical domain name

; ftp.xyz.com is an alias
; ftp.ghana.com is the canonical name
ftp.ghana.com. CNAME austin.ghana.com

Free form text

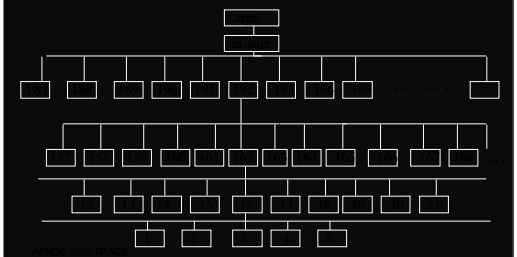
- TXT record
- Owner is any domain name
- Data is any text associated with the domain name
- Very few conventions about how to use it

net.ghana.com. TXT "NETWORKS R US"

Reverse Lookup

- When a source host establishes a connection to a destination host, the TCP/IP packets carry out only IP addresses of the source host;
- For authentication, access rights or accounting information, the destination host wants to know the name of the source host;
- For this purpose, a special domain "in-addr.arpa" is used;
- The reverse name is obtained by reversing the IP number and adding the name "in-addr.arpa";
- Example: address: 130.65.240.254
  reverse name: 254.240.65.130.in-addr.arpa
- Reverse domains form a hierarchical tree and are treated as any other Internet domain.
- Rfc2317 Classless In-ADDR.ARPA delegation

Reverse Domain Hierarchy



Requirements for a nameserver

- A query should be resolved as fast as possible;
- It should be available 24 hours a day;
- It should be reachable via fast communication lines;
- It should be located in the central in the network topology;
- It should run robust, without errors and interrupts.

How is data partitioned amongst the servers?

- The namespace is divided into zones
- Each zone has two or more authoritative nameservers
  - One primary or master
  - One or more secondaries or slaves
  - Slaves periodically update from master
- Each server is authoritative for any number of zones (zero or more)

DNS Protocols

- Client/server question/answer
  - What kinds of questions can clients ask?
  - The resolver/server model
  - What if the server does not know the answer?
- Master and slave servers
  - Configuration by zone administrator
  - Periodic update of slaves from master

What kinds of questions can clients ask?

- All the records of a particular type for a particular domain name
  - All the A records, or all the MX records
- All records of any type for a particular domain name
- A complete zone transfer of all records in a particular zone
  - Used to synchronise slave with master server

What if the server does not know the answer?

- Servers that receive queries for which they have no information can return a referral to another server
- Referral may include SOA, NS records and A records
- Client can recursively follow the referral
- Server may recurse on behalf of client, if client so requests and server is willing

Master and slave servers

- a.k.a. primary and secondary
- zone administrator sets up primary/master
- asks friends or ISPs to set up slaves/secondaries
- slave periodically checks with master to see if data has changed
- transfers new zone if necessary
- serial number in SOA record in each zone

Location of servers

- one master and at least one slave
- on different networks
- avoid having a single point of failure
- RFC 2182- SELECTION AND OPERATION OF SECONDARY DNS SERVERS
- RFC2181- CLARIFICATIONS TO THE DNS SPECIFICATION

Configuring a resolver on a Unix-like system

- Unix-like systems use /etc/resolv.conf file
- resolver is part of libc or libresolv, compiled into application programs
- resolv.conf says which nameservers should be used by the resolver
- resolv.conf also has other functions, see the resolver or resolv.conf man pages

resolv.conf example

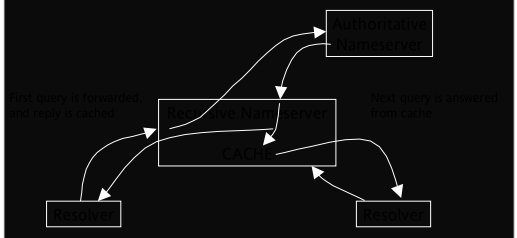
- /etc/resolv.conf file contains the following lines

```
domain  t1.ws.afnog.org
nameserver 80.248.72.100
nameserver 80.248.72.254
```


The resolver/server model

- user software asks resolver a question
- resolver asks server
- server gives answer, error, or referral to a set of other servers
- server may recurse, or expect resolver to recurse
- caching
- authoritative/non-authoritative answers

The resolver/server model (diagram)



Configuring a nameserver on a Unix-like system

- BIND is the most common implementation
- up to version 4.9.* use /etc/named.boot file
- from version 8.* use /etc/named.conf file
- cache name
- primary/master zone name and file name
- secondary/slave zone name, master IP address, backup file name

named.boot example

- /etc/named.boot contains the following lines

```
directory /etc/namedb
; type      zone              master      file name
cache       .                             root.cache
primary     t1.ws.afnog.org               afnog.org
secondary gh.com  196.3.64.1 sec/gh.com
```


named.conf example

- /etc/named.conf contains the following lines

```
options { directory "/etc/namedb"; };
zone "." { type ; file "root.cache"; };
zone "t1.ws.afnog.org" { type master; file
  "afnog.org"; };
zone "gh.com" { type slave; masters { 196.3.64.1;
  }; file "sec/gh.com"; };
```


Checking DNS using nslookup

- nslookup commands:

```
server <nameserver>       ; set the server to be queried
set type = NS             ;queries NS resources
set type = SOA            ;queries SOA resources
set type = A              ;queries A resources
set type = MX             ;queries MX resources
set type = CNAME          ;queries CNAME resources
set type = PTR            ;queries PTR resources
set type = ANY            ;queries ANY resources
ls <domain>               ;lists the <domain>. zone
ls <domain> > <file-name> ;gets the zone <domain> into the
                           file=file-name>
```


Checking DNS using dig

- Dig
  - Tool to manage DNS settings
  - Syntax is:

```
dig [domain] @nameserver [query-type]
```

Best Practices

- Upgrade to latest version of BIND
- Always Increment your serial number
- Inform hostmasters of orgs you to run name service for you.
- MX servers should know about your domain otherwise mail bounces.
- Always signal to reload after making changes

Best Practices

- Don't forget to add reverse delegation
- make sure you don't have syntax errors in conf file and zone files
- don't forget to add trailing dots in database file
- Proper Subdomain delegation
  - missing subdomain delegation
  - incorrect subdomain delegation

Best Practices

- Syntax error in resolv.conf
- don't forget to set your default domain

Checking for DNS correctness

- Several Programs available
- ftp://ftp.isc.org/isc/bind/src/8.1.1/bind-contrib.tar.gz
- www.domtools.com

Questions