The Domain Name System (DNS)

Raj Jain

Raj Jain is now at
Washington University in Saint Louis
Jain@cse.wustl.edu
http://www.cse.wustl.edu/~jain/
Overview

- Naming hierarchy
- Server hierarchy
- Name resolution
- Other information in name servers
Why Names?

- Computers use addresses
- Humans cannot remember IP addresses ⇒ Need names
  - Example, Liberia for 164.107.51.28
- Simplest Solution: Each computer has a unique name and has a built-in table of name to address translation
- Problem: Not scalable
- Solution: DNS (Adopted in 1983)
- Hierarchical Names: Liberia.cis.ohio-state.edu
Name Hierarchy

Unnamed root

com

dec

cis

edu

netlab

ohio-state

gov

nsf

au

co

au...

us

va

reston

cnri
Name Hierarchy

- Unique domain suffix is assigned by Internet Authority
- The domain administrator has complete control over the domain
- No limit on number of subdomains or number of levels
- computer.site.division.company.com
- computer.site.subdivision.division.company.com
- Domains within an organization do not have to be uniform in number of subdomains or levels
Name Hierarchy (Cont)

- Name space is not related to physical interconnection, e.g., math.ohio-state and cis.ohio-state could be on the same floor or in different cities.

- Geographical hierarchy is also allowed, e.g., cnri.reston.va.us.

- A name could be a subdomain or an individual object.
### Top Level Domains

<table>
<thead>
<tr>
<th>Domain Name</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>com</td>
<td>Commercial</td>
</tr>
<tr>
<td>edu</td>
<td>Educational</td>
</tr>
<tr>
<td>gov</td>
<td>Government</td>
</tr>
<tr>
<td>mil</td>
<td>Military</td>
</tr>
<tr>
<td>net</td>
<td>Network</td>
</tr>
<tr>
<td>org</td>
<td>Other organizations</td>
</tr>
<tr>
<td>arpa</td>
<td>Advanced Research Project Agency</td>
</tr>
<tr>
<td>country code</td>
<td>au, uk, ca</td>
</tr>
</tbody>
</table>
Server Hierarchy
Server Hierarchy (Cont)

- Servers are organized in a hierarchy
- Each server has an authority over a part of the naming hierarchy
- The server does not need to keep all names.
- It needs to know other servers who are responsible for other subdomains
- Contiguous space \(\Rightarrow\) A single node in the naming tree cannot be split
- A given level of hierarchy can be partitioned into multiple servers
Server Hierarchy (Cont)

- Authority ⇒ has the name to address translation table
- Responsible ⇒ Either has the name to address translation table or knows the server who has
- A single server can serve multiple domains, e.g., purdue.edu and laf.in.us
- Root server knows about servers for top-level domains, e.g., com
- Each server knows the root server
Server Hierarchy: Example

Root Server

- Server for com
  - Server for dec.com
- Server for edu
  - Server for osu.edu
- Server for gov
  - Server for nsf.gov
- Server for au
  - Server for co.au
- Server for us
  - Server for va.us
Server Hierarchy: Better

- Fewer servers

Root Server

- Server for dec.com
- Server for osu.edu
- Server for nsf.gov
- server for co.au
- Server for va.us
Name Resolution

User

Name Server

Cache

Data base

Query

Response

Name Server

Cache

Data base

Query

Response

Name Resolver

Cache

User

Query

Response

Name Server

Cache

Data base

Query

Response

Name Server

Cache
Name Resolution (Cont)

User Name

Name Server

Data- base

Cache

Query

Response

Name Resolver

Cache

Query

Response

Name Server

Cache

Query

Referral

Data- base

Database

Database

The Ohio State University

Raj Jain
Name Resolution (Cont)

- Each computer has a name resolver routine, e.g., gethostbyname in UNIX
- Each resolver knows the name of a local DNS server
- Resolver sends a DNS request to the server
- DNS server either gives the answer, forwards the request to another server, or gives a referral
- Referral = Next server to whom request should be sent
Name Resolution (Cont)

- Resolvers use UDP (single name) or TCP (whole group of names)
- Knowing the address of the root server is sufficient
- Recursive Query: Give me an answer (Don't give me a referral)
- Iterative Query: Give me an answer or a referral to the next server
- Resolvers use recursive query.
- Servers use iterative query.
DNS Optimization

- **Spatial Locality**: Local computers referenced more often than remote.
- **Temporal Locality**: Same set of domains referenced repeatedly $\Rightarrow$ Caching.
- Each entry has a time to live (TTL).
- **Replication**: Multiple servers. Multiple roots. Ask the geographically closest server.
Abbreviations

- Servers respond to a full name only
- However, humans may specify only a partial name
- Resolvers may fill in the rest of the suffix, e.g., Liberia.cis = Liberia.cis.ohio-state.edu
- Each resolver has a list of suffixes to try
## DNS Message Format

<table>
<thead>
<tr>
<th>Identification</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Questions</td>
<td>Number of Answers</td>
</tr>
<tr>
<td>Number of Authority</td>
<td>Number of Additional</td>
</tr>
</tbody>
</table>

### Question Section

...  

### Answer Section

...  

### Authority Section

...  

### Additional Information Section

...
Format (Cont)

- Format of the query section entries:

<table>
<thead>
<tr>
<th>Query Domain Name</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Type</td>
<td>Query Class</td>
</tr>
</tbody>
</table>

- Format of other section entries:

<table>
<thead>
<tr>
<th>Resource Domain Name</th>
<th>Resource Data Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Class</td>
</tr>
<tr>
<td>Time to live</td>
<td>Resource Data Data Length</td>
</tr>
<tr>
<td>Resource Data</td>
<td></td>
</tr>
</tbody>
</table>
DNS Message Format

- Length = 0 ⇒ End of names. Length < 64
  Two msbs (most significant bits) = 11 ⇒ Pointer
- Resource data contains serial (version) number of the zone, refresh interval, retry interval, expiry interval, mailbox of the responsible person, etc.
<table>
<thead>
<tr>
<th>Bit</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Operation: 0=Query, 1=Response</td>
</tr>
<tr>
<td>1-4</td>
<td>Query type: 0=Standard, 1=Inverse, 2,3 obsolete</td>
</tr>
<tr>
<td>5</td>
<td>Set if answer authoritative</td>
</tr>
<tr>
<td>6</td>
<td>Set if message truncated</td>
</tr>
<tr>
<td>7</td>
<td>Set if recursion desired</td>
</tr>
<tr>
<td>8</td>
<td>Set if recursion available</td>
</tr>
<tr>
<td>9-11</td>
<td>Reserved</td>
</tr>
<tr>
<td>12-15</td>
<td>Response type: 0=No error, 1=Format error, 2=Server Failure, 3=Name does not exist</td>
</tr>
</tbody>
</table>
Inverse Mapping

- Given an address, what is the name?
- nnn.nnn.nnn.nnn.in-addr.arpa
Types of DNS Entries

- DNS is used not just for name to address resolution
- But also for finding mail server, pop server, responsible person, etc for a computer
- DNS database has multiple types
- Record type A ⇒ Address of X
- Record type MX ⇒ Mail exchanger of X
- CNAME entry = Alias name (like a file link), "see name"
- www.foobar.com = hobbes.foobar.com
## Resource Record Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Host Address</td>
</tr>
<tr>
<td>CNAME</td>
<td>Canonical Name (alias)</td>
</tr>
<tr>
<td>HINFO</td>
<td>CPU and O/S</td>
</tr>
<tr>
<td>MINFO</td>
<td>Mailbox Info</td>
</tr>
<tr>
<td>MX</td>
<td>Mail Exchanger</td>
</tr>
<tr>
<td>NS</td>
<td>Authoritative name server for a domain</td>
</tr>
<tr>
<td>PTR</td>
<td>Pointer to a domain name (link)</td>
</tr>
<tr>
<td>RP</td>
<td>Responsible person</td>
</tr>
<tr>
<td>SOA</td>
<td>Start of zone authority (Which part of naming hierarchy implemented)</td>
</tr>
<tr>
<td>TXT</td>
<td>Arbitrary Text</td>
</tr>
</tbody>
</table>
Summary

- DNS: Maps names to addresses
- Names are hierarchical. Administration is also hierarchical.
- No standard for number of levels
- Replication and caching is used for performance optimization.