Domain Name System

Richard T. B. Ma
School of Computing
National University of Singapore

CS 3103: Compute Networks and Protocols
Names Vs. Addresses

- Names are easier for humans to remember
  - www.comp.nus.edu.sg vs. 137.132.90.2

- Addresses can be changed without changing names
  - move www.comp.nus.edu.sg to 203.126.100.199

- Name could map to multiple addresses
  - google.com maps to multiple replicas of the Web site and to different “nearby” addresses in different geographies to reduce latency

- Multiple names could map to the same address
The Domain Name System (DNS) is a distributed database implemented in a hierarchy of DNS servers, and an application-layer protocol that allows hosts to query the distributed database.

The DNS servers are often UNIX machines running the Berkeley Internet Name Domain (BIND) software.

The DNS protocol runs over UDP on port 53.

Used by other application-layer protocols:
- E.g., HTTP, SMTP, and FTP
A Typical Scenario

User

Host name

File transfer client

DNS client

DNS server

Query

Response

Transport layer

Application layer

Host name

IP address

IP address
DNS: Other Use Cases

- **Host aliasing**
  - Canonical hostname \(\rightarrow\) alias hostname
  - E.g., webserver1.abc.com \(\rightarrow\) www.abc.com

- **Mail server aliasing**
  - Bob’s account with Hotmail \(\rightarrow\) bob@hotmail.com
  - Web and mail servers share the same

- **Load distribution**
  - Load balancing among replicated servers
  - Rotate among the set of IP addresses in replies
Hierarchical Name Space

- Maximum 128 levels (level 0 to level 127)
- The label for each level is a string with maximum of 63 characters
- The root label is a null string
Hierarchical Name Space

- **Name allocation decentralized to domains**
  
  ```
  host.sub-subdomain...subdomain.domain[.ROOT]
  ```
  
  - **host**: machine name, can be an alias
  - **sub-subdomain**: department (comp, eng, ceg, math)
  - **subdomain**: institution, company, geography, provider
  - **domain**: most significant segment (edu, com, org, net, gov)

- **Domain**: a sub-tree of the domain name space

- **Domain name**
  - Fully Qualified Domain Name (FQDN)
  - Partially Qualified Domain Name (PQDN)
Implementation

- Distributed database implemented in hierarchy of many name servers
  - A distributed database storing resource records (RR)
  - Client-server query-reply

- Host, routers and name servers communicate to resolve domain names
  - core Internet function, implemented as application-layer protocol
  - complexity at network’s “edge”
client wants IP for www.amazon.com; 1st approx:

- client queries a root server to find com DNS server
- client queries com DNS server to get amazon.com DNS server
- client queries amazon.com DNS server to get IP address for www.amazon.com
Discussion: Design Choices

- Why not flat name space?

- Why not centralize DNS?
  - Single point of failure
  - Traffic volume
  - Distant centralized database
  - Maintenance

Scalability!!
DNS: Root name servers

- contacted by local name server that can not resolve name
- root name server:
  - contacts authoritative name server if name mapping not known
  - gets mapping
  - returns mapping to local name server

13 root name servers worldwide
TLD and Authoritative Servers

Top-level domain (TLD) servers:

- responsible for com, org, net, edu, aero, jobs, museums, and all top-level country domains, e.g.: uk, fr, ca, jp
- Network Solutions maintains servers for com TLD
- Educause for edu TLD

Authoritative DNS servers:

- organization’s DNS servers, providing authoritative hostname to IP mappings for organization’s servers (e.g., Web, mail).
- can be maintained by organization or service provider
Local Name Server

- Does not strictly belong to hierarchy
  - Like a “client” (the hierarchy is the “server”)

- Each ISP (residential ISP, company, university) has one
  - also called “default name server”

- When host makes DNS query, query is sent to its local DNS server
  - acts as proxy, forwards query into hierarchy
DNS name resolution example

- Host at cis.poly.edu wants IP address for gaia.cs.umass.edu

**iterated query:**
- contacted server replies with name of server to contact
- “I don’t know this name, but ask this server”
**DNS name resolution example**

**recursive query:**
- puts burden of name resolution on contacted name server
- heavy load?

![Diagram of DNS name resolution example](image_url)
DNS: caching and updating records

- Once (any) name server learns mapping, it **caches** the mapping
  - cache entries timeout (disappear) after some time
  - TLD servers typically cached in local name servers
    - Thus root name servers not often visited

- Update/notify mechanisms proposed IETF standard
  - RFC 2136
DNS records

**DNS**: distributed db storing resource records (RR)

RR format: \((name, \text{value}, \text{type}, \text{ttl})\)

**Type=NS (Name Server)**
- name is domain (e.g., foo.com)
- value is hostname of authoritative name server for this domain

**Type=A (Address)**
- name is hostname
- value is IP address

**Type=CNAME (Canonical NAME)**
- name is alias name for some “canonical” (the real) name
- value is canonical name

**Type=MX (Mail eXchange)**
- value is name of mailserver associated with name

- example: www.ibm.com is really servereast.backup2.ibm.com
DNS protocol, messages

- **query and reply** messages, both with same message format

msg header
- **identification**: 16 bit # for query, reply to query uses same #
- **flags:**
  - query or reply
  - recursion desired
  - recursion available
  - reply is authoritative

<table>
<thead>
<tr>
<th></th>
<th>identification</th>
<th>flags</th>
</tr>
</thead>
<tbody>
<tr>
<td># questions</td>
<td></td>
<td># answer RRs</td>
</tr>
<tr>
<td># authority RRs</td>
<td></td>
<td># additional RRs</td>
</tr>
<tr>
<td>questions (variable # of questions)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>answers (variable # of RRs)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>authority (variable # of RRs)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>additional info (variable # of RRs)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### DNS protocol, messages

<table>
<thead>
<tr>
<th>Identification</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td># questions</td>
<td># answer RRs</td>
</tr>
<tr>
<td># authority RRs</td>
<td># additional RRs</td>
</tr>
</tbody>
</table>

- **identification**
- **flags**
- **# questions**
- **# answer RRs**
- **# authority RRs**
- **# additional RRs**

- **questions (variable # of questions)**
- **answers (variable # of RRs)**
- **authority (variable # of RRs)**
- **additional info (variable # of RRs)**

- **name, type fields for a query**
- **RRs in response to query**
- **records for authoritative servers**
- **additional “helpful” info that may be used**
Inserting records into DNS

- example: new startup “Network Utopia”
- register name networkuptopia.com at DNS registrar (e.g., Network Solutions)
  - provide names, IP addresses of authoritative name server (primary and secondary)
  - registrar inserts two RRs into com TLD server:
    - (networkutopia.com, dns1.networkutopia.com, NS)
    - (dns1.networkutopia.com, 212.212.212.1, A)

- create authoritative server Type A record for www.networkuptopia.com; Type MX record for networkutopia.com
- How do people get IP address of your Web site?