Domain Name System

• Map between host names and IP addresses
  • **People:** many identifiers: name, Passport #, ...
  • **Internet hosts:**
    - IP address (32 bit) - used for addressing datagrams
    - “name”, e.g., www.iitb.ac.in - used by humans
DNS

• Provides logical hierarchical view of the Internet
  • globally distributed database implemented in hierarchy of many name servers
  • application-layer protocol to communicate to resolve names (address/name translation)
  • client/server interaction
DNS clients and servers

- **clients**: query servers to resolve names; nslookup
- **servers**: name server daemons, reply to queries; BIND, named
- **gethostbyname**: resolver library call that can be invoked from application program

Lazily validated cache for performance
DNS design

Centralized DNS?
- single point of failure
- traffic volume
- distant centralized database
- maintenance
- doesn’t scale!

So…Distributed DNS
Name hierarchy

- Unique domain suffix is assigned by Internet Authority
- No limit on number of subdomains or number of levels
- Domains within an organization do not have to be uniform in number of subdomains or levels
  - www.iitb.ac.in
  - www.it.iitb.ac.in
# 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>
DNS hierarchy

- 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
DNS: Local name servers

Local Name Servers:

- each organization/ISP has *local (default) name server*
- host DNS query first goes to local name server

Authoritative Name Server:

- for a host: stores that host’s IP address, name
- can perform name/address translation for that host’s name
Name server hierarchy

- A single server can serve multiple domains
- Root server knows about servers for top-level domains
- Each server knows the root server
DNS: Root name servers

- Contacted by local name server that cannot resolve name
- Root Name Server:
  - contacts authoritative name server if name mapping not known
  - gets mapping
  - returns mapping to local name server

Several root name servers worldwide
DNS hierarchy: Example

Root Server

- Server for com
  - Server for ibm.com
- Server for edu
  - Server for rpi.edu
- Server for gov
  - Server for nsf.gov
- Server for in
  - Server for co.in
- Server for us
  - Server for va.us
DNS: Example

host xyz.iitb.ac.in wants IP address of www.ibm.com

1. Contacts its local DNS server, dns.iitb.ernet.in
2. dns.iitb.ernet.in contacts root name server, if necessary
3. root name server contacts authoritative name server, dns.ibm.com, if necessary
DNS: Name resolution

- Recursive queries:
  - puts burden of name resolution on contacted name server
  - not scalable under heavy load

- Iterated queries:
  - contacted server replies with name of server to contact. Ex: root name server may know *intermediate name server* to contact to find authoritative name server
Recursive queries

Diagram:
- User
- Name Server
- Name Resolver
- Database
- Cache

Flow:
1. User queries Name Resolver
2. Name Resolver queries Name Server
3. Name Server provides response to Name Resolver
4. Name Resolver provides response to User
Iterated queries
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.
DNS: caching and updating

• A name server *caches* the mappings learnt
  • cache entries have a time-to-live period after which they become invalid
• update/notify mechanisms: RFC 2136
DNS record

- **Resource Record (RR) format**: (name, value, type, ttl)
- **Type=A**: name is hostname; value is IP address
- **Type=NS**: name is domain (e.g. ibm.com); value is IP address of authoritative name server for this domain
- **Type=CNAME**: name is an alias name for some “cannonical” (the real) name; value is cannonical name
- **Type=MX**: value is hostname of mailserver associated with name
DNS protocol

- client-server interaction
  - query and reply messages, both with same message format

- Message header
  - identification: 16 bit # for query, reply uses same #
  - flags: query or reply; recursion desired; recursion available; reply is authoritative
## DNS message format

<table>
<thead>
<tr>
<th>Identification</th>
<th>Flags</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>

<table>
<thead>
<tr>
<th>Question Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Answer Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Authority Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Additional Information Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
</tr>
</tbody>
</table>