The Domain Name System

The 411 of the Internet
Host names

- Internet hosts may be identified by a unique hostname.

- Similar to telephone network, the hostname consists of a name (resource name) plus an address (domain name).

- Hostnames are good for people, but the Internet uses 32 bit IP addresses for sending datagrams.

IP addresses*

- Consists of four bytes expressed in decimal notation from 0 to 255

- Hierarchical structure: scan from left to right yields more specific information about host's location

*You guessed it, more later when we get to the network layer.
Domain Name System (DNS)

- The Domain Name System is a distributed database implemented in a hierarchy of name servers.

- It is also an application-layer protocol that provides 411 service between hosts and name servers.

- All our old friends use DNS including HTTP, SMTP, and FTP.

HTTP uses Internet 411


2. DNS server does its thing and IP address is sent to DNS client.

3. DNS client receives reply and passes IP address to HTTP client so that browser can open a TCP connection to HTTP server.

4. HTTP server receives request and the rest is history.
HTTP uses Internet 411


2. DNS server does its thing and IP address is sent to DNS client.

3. DNS client receives reply and passes IP address to HTTP client so that browser can open a TCP connection to HTTP server.

4. HTTP server receives request and the rest is history.

Both HTTP and DNS clients live on the same host

HTTP and DNS servers live on different hosts

A number of hosts may get involved here, introducing additional (sometimes substantial) delay
Wouldn’t a centralized design be simpler?

- Well, it would be a single point of failure.
- Traffic to and from a single server would be worse than Boston at 8:00 am on a Monday morning.
- A single name server would not be close to everyone. Someone in either New York City or Sidney, Australia would not be happy.
- Maintenance of a centralized database would be a nightmare.

Distributed, hierarchical database

- Top-level domain (TLD) servers
  - com DNS servers
    - yahoo.com DNS servers
  - org DNS servers
    - amazon.com DNS servers
  - edu DNS servers
    - pbs.org DNS servers
    - poly.edu DNS servers
    - wellesley.edu DNS servers
- Root DNS Servers
  - Root servers
- Authoritative servers
  - cheers.wellesley.edu
  - The cheese stands alone
The name server hierarchy

Each ISP has a local name server where the client DNS request is sent first.

List is generated dynamically at startup and stored in /etc/resolv.conf

One of two authoritative name servers for wellesley.edu domain*

Windows Active Directory Domain Controllers

*From inside our domain; the outward facing DNS servers have been outsourced.

Root name servers

- If Cheers doesn’t know the IP address, she behaves as a DNS client and queries one of the root name servers.

Each server is actually a cluster of replicated servers, for both security and reliability purposes.
Passing the buck

• Top-level domain servers are responsible for top-level domains (com, org, net, edu, and gov).

• The root server will probably refer us to one of these.*

*You see where this is going.

Authoritative name servers

• All within a domain hosts are registered with an authoritative name server* which is where we will probably be referred to next.

• The authoritative name server for a host is generally in the hosts local ISP.
cis.poly.edu looks up gaia.cs.umass.edu

After the dance, the IP address would be cached locally for future reference.

But what if …

- … the TLD DNS server doesn’t know the authoritative DNS server for gaia.cs.umass.edu.

- Perhaps U Mass has a name server for the entire university, called dns.umass.edu, …

- … and each department has its own name server that is authoritative for all hosts in the department.
Busy servers

• **Recursive query**: Server A makes a request on behalf of server B, then forwards the address.

• This puts the burden of name resolution on the contacted server.

• Root servers have places to go, people to meet.

Iterated queries

• **Iterated query**:
  – Server A queries server B.
  – If Server B doesn’t know, she responds with the name of the next server in the chain.

• In effect, server B says, “I don’t know this name. Go ask so-and-so.”
DNS caching

- In a query chain, when a DNS server receives a DNS reply it caches the mapping in its local memory.
- When a second query arrives for the same hostname, it can provide the IP address, even if it is not authoritative for the hostname.
- DNS servers typically discard cached information after a couple of days.

DNS resource records

- **TTL** is the time to live.
- **Name** and **Value** depend on type.
- **Type=A**
  - Name is hostname and Value is IP address
- **Type=NS**
  - Name is domain and Value is the hostname of an authoritative server that knows how to obtain IP addresses for hosts in the domain
- **Type=CNAME**
  - Name is alias for some canonical name and Value is the canonical name.
- **Type=MX**
  - Value is the name of mailserver associated with Name.
**DNS protocol, messages**

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

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>identification</strong></td>
<td><strong>flags</strong></td>
</tr>
<tr>
<td><strong># questions</strong></td>
<td><strong># answer RRs</strong></td>
</tr>
<tr>
<td><strong># authority RRs</strong></td>
<td><strong># additional RRs</strong></td>
</tr>
</tbody>
</table>

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

**msg header**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

```
2 bytes  2 bytes
```

```sql
21
```
How do records get into the database in the first place?

- To create a domain, we first
  - register the domain name,
    - say mynewdomain.edu
  - with a commercial entity known as a registrar
    - including the IP addresses of our primary and second authoritative servers.

- The registrar would then enter a type NS and type A records into the TLD servers for each name server.

- For example, suppose your primary name servers was named dns1.mynewdomain.edu:
  - (mynewdomain.edu, dns1.mynewdomain.edu, NS)
  - (dns1.mynewdomain.edu, 212.212.212.1, A)

Scary times

- To say that DNS is a important component of the Internet infrastructure is putting it mildly.

- Without it many important services (Web, email, ...) would cease to function.

- How vulnerable is DNS to attack? And who would want to attack it?
DDoS bandwidth-flooding attack

- A large-scale DDoS attack against DNS root servers was launched on October 21, 2002.

- The attackers leveraged a botnet to send truck loads of ICMP ping messages to each of the 13 DNS root servers.

- Fortunately, the root servers were protected by packet filters, configured to block all ICMP ping messages.

- A more effective DDoS attack would send a deluge of DNS queries to top-level-domain servers.

Man-in-the-middle attack

- In a man-in-the-middle attack, the attacker intercepts queries from hosts and returns bogus replies.

- In the DNS poisoning attack, the attacker sends bogus replies to a DNS server, tricking the server into accepting bogus records into its cache.

- Either attack could be used to redirect an unsuspecting Web user to the attacker's website.