Recap SMTP and email

- SMTP process (with handshaking) and message format
- Role of user agent access protocols
- Port Numbers (can google this)
  - SMTP - 25
  - POP - 110
  - IMAP - 143
  - HTTP - 80
  - (DNS - 53 over UDP)

Scenario: Alice sends message to Bob

Sample SMTP interaction

- In the following interaction with SMTP, which lines are
  - Handshaking
  - Transfer of message
  - Closure
Sample SMTP interaction

```
fcapmaster:~ jcardell$ telnet smtp.smith.edu 25
Trying 131.229.64.236...
Connected to baton.smith.edu.
Escape character is '^]'.
220 baton.smith.edu ESMTP Sendmail 8.13.8/8.13.8; Tue, 16 Sep 2014 11:37:00 -0400

C: HELO jbc.edu
S: 250 baton.smith.edu Hello [131.229.102.128], pleased to meet you
C: MAIL FROM: <judy@jbc.edu>
S: 250 2.1.0 <judy@jbc.edu>... Sender ok
C: RCPT TO: <jcardell@smith.edu>
S: 250 2.1.5 <jcardell@smith.edu>... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: to: Easter@Bunny.hop
C: from: Tooth@Fairy.fly
C: subject: How's Business?
C: Hello Hoppy
C: Checking in to see how you're doing.
C: .
S: 250 2.0.0 s8GFb0Q4007216 Message accepted for delivery
C: QUIT
S: 221 2.0.0 baton.smith.edu closing connection
```
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 & 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 own DNS server(s), providing authoritative hostname to IP mappings for organization’s named hosts
- can be maintained by organization or service provider

Local DNS name server

- does not strictly belong to hierarchy
- 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
  - has local cache of recent name-to-address translation pairs (but may be out of date)
  - acts as proxy, forwards query into hierarchy
  - When you connect to network, your host is given the IP address of the local DNS server

DNS name resolution example

- host at www.smith.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 protocol, messages

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

  **Message 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>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>
<tr>
<td>questions (variable # of questions)</td>
<td>answers (variable # of RRs)</td>
</tr>
<tr>
<td>authority (variable # of RRs)</td>
<td>additional info (variable # of RRs)</td>
</tr>
</tbody>
</table>

### HTTP request message: general format

<table>
<thead>
<tr>
<th>Method</th>
<th>URL</th>
<th>sp</th>
<th>Version</th>
<th>cr</th>
<th>if</th>
<th>Request line</th>
</tr>
</thead>
<tbody>
<tr>
<td>header field name</td>
<td>value</td>
<td>cr</td>
<td>if</td>
<td>header lines</td>
<td></td>
<td></td>
</tr>
<tr>
<td>header field name</td>
<td>value</td>
<td>cr</td>
<td>if</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Entity Body</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Mail message format

- Example of the actual message - NOT part of the SMTP handshaking process
  - **header** lines, e.g.,
    - To:
    - From:
    - Subject: different from SMTP commands!
  - **body**
    - the “message”, ASCII characters only
**Domain Name System protocol messages**

- Name, type fields for a query
- RR in response to query
- Records for authoritative servers
- Additional "helpful" info that may be used

**DNS record format**

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

**RR format:** (name, value, type, ttl)

- **Type=A**
  - name is hostname
  - value is IP address
- **Type=NS**
  - name is domain (e.g. smith.edu)
  - value is hostname of authoritative name server for this domain
- **Type=CNAME**
  - name is alias name for some "canonical" (the real) name
  - value is canonical name
- **Type=MX**
  - value is name of mailserver associated with name

**DNS records**

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

**RR format:** (name, value, type, ttl)

- (hostname, IP address, A, ttl)
- (domain, hostname-DNS-author-server, NS, ttl)
- (alias hostname, canonical name, CNAME, ttl)
- (alias hostname, mail server cname, MX, ttl)

**DNS Records**

1. Requesting Host
2. root DNS server
3. TLD DNS server
4. + Type NS record with hostname of auth. server for the requested domain name
5. + and Type A record for IP address of auth. server
6. authoritative DNS server
7. TLD DNS server
8. + Type A record for IP address of 'hostname'
9. local DNS server
10. Requesting Host
11. authoritative DNS server
gail.cs.umass.edu
12. dns.cs.umass.edu
13. TLD DNS server
* Act out the DNS process *

**DNS protocol**: query and reply messages, both with same message format

**Message header**
- **identification**: 16 bit # for query, reply to query uses same #
- **flags**
- **Number of records in the message itself**

<table>
<thead>
<tr>
<th>Identification</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of questions</td>
<td>number of answer RR's</td>
</tr>
<tr>
<td>number of authority RR's</td>
<td>number of additional RR's</td>
</tr>
</tbody>
</table>

### nslookup with Mac OS

![nslookup with Mac OS screenshot](image)

**Summary of Application Design Elements**

- **Message format**
  - ASCII? Binary?
  - How handle multiple objects?
- **Number of connections**
  - Persistent? Parallel connections?
- **State information? Stateless?**
- **TCP or UDP (Transport Layer)?**
- **Push or pull protocol?**
- **How to find the server? client? peer?**
- **Handshaking in the protocol?**
- **Centralized? Decentralized?**