**DNS: Domain Name System**
- People have many identifiers:
  - SSN, name, passport number
- Internet hosts, routers have identifiers too:
  - IP address (32 bit) - used for addressing datagrams
  - "name", e.g., www.yahoo.com - used by humans
- Domain Name System:
  - Distributed database implemented in hierarchy of many server names
  - Application-layer protocol host, routers, name servers to communicate to resolve names (address/name translation)
  - DNS is core Internet function, implemented as application-layer protocol
  - Complexity at network's "edge"

**DNS**
- DNS services
  - Hostname to IP address translation and vice versa
  - Host aliasing - canonical and alias names
  - Mail server aliasing
  - Load distribution - e.g. replicated Web servers
  - Set of IP addresses for one canonical name
- Why not centralize DNS?
  - Single point of failure
  - Traffic volume
  - Distant centralized database
  - Maintenance

**Distributed, Hierarchical Database**
- Distributed DNS servers
  - .com DNS servers
  - .org DNS servers
  - .edu DNS servers
  - yahoo.com DNS servers
  - amazon.com DNS servers
  - pbs.org DNS servers
  - poly.edu DNS servers
  - umass.edu DNS servers
  - DNS servers

- Client wants IP for www.amazon.com
  - A straightforward strategy:
    - 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

**DNS: Root name servers**
- Contacted by local name server that cannot resolve name
- Root name server:
  - Returns mapping of the domain server to local name server

**TLD and Authoritative Servers**
- Top-level domain (TLD) servers:
  - Responsible for .com, .org, .net, .edu, etc. and all top-level country domains .uk, .fr, .ca, .jp.
  - Network solutions maintains servers for .com TLD
  - Education for .edu TLD
- Authoritative DNS servers:
  - Organization's DNS servers, providing authoritative hostname to IP mappings for organization's servers (e.g., Web and mail)
  - Can be maintained by organization or service provider

**Local Name Server**
- Does not strictly belong to DNS hierarchy
- Each ISP (residential ISP, company, university) has one
  - Also called "default name server"
- When a host makes a DNS query, query is sent to its local DNS server
  - Acts as a cache
  - If no record for this query, forwards query into hierarchy
  - Caches replies
Example:

Host titan.cis.udel.edu wants IP address for lever.cs.ucla.edu

Recursive queries:
- Put burden of name resolution on contacted name server
- Heavy load?
- Iterated query:
  - Contacted server replies with name of server to contact
  - "I don’t know this name, but ask this server"

DNS: caching and updating records
- Once (any) name server learns mapping, it caches this mapping:
  - TLD entries timeout (disappear) after some time
  - TLD servers typically cached in local name servers
    - Thus root name servers are not often visited
- Update/notify mechanisms under design by IETF
  - RFC 2136

DNS resource records (RR)

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Class</th>
<th>TTL</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>mycompany.com</td>
<td>SOA</td>
<td>IN</td>
<td>10800</td>
<td>10.0.0.100</td>
</tr>
<tr>
<td>nameserv.mycompany.com</td>
<td>NS</td>
<td>IN</td>
<td>86400</td>
<td>nameserv.mycompany.com</td>
</tr>
<tr>
<td>mail.mycompany.com</td>
<td>MX</td>
<td>IN</td>
<td>3600</td>
<td>10.0.0.100</td>
</tr>
<tr>
<td><a href="http://www.mycompany.com">www.mycompany.com</a></td>
<td>CNAME</td>
<td>IN</td>
<td>900</td>
<td>webhost1.mycompany.com</td>
</tr>
<tr>
<td>webhost1.mycompany.com</td>
<td>A</td>
<td>IN</td>
<td>300</td>
<td>10.0.1.13</td>
</tr>
</tbody>
</table>

DNS protocol, messages:

DNS protocol has query and reply messages, both with same message format:
- Identification
- Flags
- Number of Questions
- Number of Answer Records
- Number of Authority Records
- Number of Additional Records
- Query/Response Flags:
  - Query or Recursion Desired
  - Recursion Available
  - Reply is Authoritative

DNS resource records (RR)

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Class</th>
<th>TTL</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>mycompany.com</td>
<td>SOA</td>
<td>IN</td>
<td>10800</td>
<td>10.0.0.100</td>
</tr>
<tr>
<td>nameserv.mycompany.com</td>
<td>NS</td>
<td>IN</td>
<td>86400</td>
<td>nameserv.mycompany.com</td>
</tr>
<tr>
<td>mail.mycompany.com</td>
<td>MX</td>
<td>IN</td>
<td>3600</td>
<td>10.0.0.100</td>
</tr>
<tr>
<td><a href="http://www.mycompany.com">www.mycompany.com</a></td>
<td>CNAME</td>
<td>IN</td>
<td>900</td>
<td>webhost1.mycompany.com</td>
</tr>
<tr>
<td>webhost1.mycompany.com</td>
<td>A</td>
<td>IN</td>
<td>300</td>
<td>10.0.1.13</td>
</tr>
</tbody>
</table>
DNS protocol, messages

Try out DNS!

nslookup www.microsoft.com
Server: 127.0.0.1
Address: 127.0.0.1#53

Non-authoritative answer:
www.microsoft.com canonical name = solveit! www.ms.akadns.net
toggle.www.ms.akadns.net canonical name = www.ms.akadns.net
g.www.ms.akadns.net canonical name = lb1.www.ms.akadns.net
Name: lb1.www.ms.akadns.net
Address: 207.46.199.30
Name: lb1.www.ms.akadns.net
Address: 207.46.18.30
Name: lb1.www.ms.akadns.net
Address: 207.46.19.30

Try out DNS!

nslookup www.yahoo.com
Server: 127.0.0.1
Address: 127.0.0.1#53

Non-authoritative answer:
www.yahoo.com canonical name = www.yahoo.akadns.net
toggle.www.yahoo.akadns.net canonical name = www.yahoo.akadns.net
g.www.yahoo.akadns.net canonical name = lb1.www.yahoo.akadns.net
Name: lb1.www.yahoo.akadns.net
internet address = 68.142.226.53
internet address = 68.142.226.56
internet address = 68.142.226.60
internet address = 68.142.226.10
internet address = 68.142.226.14
internet address = 68.142.226.18
internet address = 68.142.226.22
internet address = 68.142.226.26
internet address = 68.142.226.30
internet address = 68.142.226.34
internet address = 68.142.226.38
internet address = 68.142.226.42
internet address = 68.142.226.46
internet address = 68.142.226.50
internet address = 68.142.226.54
internet address = 68.142.226.58
internet address = 68.142.226.62
internet address = 68.142.226.66
internet address = 68.142.226.70
internet address = 68.142.226.74
internet address = 68.142.226.78
internet address = 68.142.226.82
internet address = 68.142.226.86
internet address = 68.142.226.90
internet address = 68.142.226.94
internet address = 68.142.226.98
internet address = 68.142.226.102

Try out DNS!

nslookup -debug www.yahoo.com

QUESTIONS:
- www.yahoo.com, type = A, class = IN
- www.yahoo.com canonical name = www.yahoo.akadns.net
- www.yahoo.com internet address = 68.142.226.53
- www.yahoo.com internet address = 68.142.226.56
- www.yahoo.com internet address = 68.142.226.60
- www.yahoo.com internet address = 68.142.226.10
- www.yahoo.com internet address = 68.142.226.14
- www.yahoo.com internet address = 68.142.226.18
- www.yahoo.com internet address = 68.142.226.22
- www.yahoo.com internet address = 68.142.226.26
- www.yahoo.com internet address = 68.142.226.30
- www.yahoo.com internet address = 68.142.226.34
- www.yahoo.com internet address = 68.142.226.38
- www.yahoo.com internet address = 68.142.226.42
- www.yahoo.com internet address = 68.142.226.46
- www.yahoo.com internet address = 68.142.226.50
- www.yahoo.com internet address = 68.142.226.54
- www.yahoo.com internet address = 68.142.226.58
- www.yahoo.com internet address = 68.142.226.62
- www.yahoo.com internet address = 68.142.226.66
- www.yahoo.com internet address = 68.142.226.70
- www.yahoo.com internet address = 68.142.226.74
- www.yahoo.com internet address = 68.142.226.78
- www.yahoo.com internet address = 68.142.226.82
- www.yahoo.com internet address = 68.142.226.86
- www.yahoo.com internet address = 68.142.226.90
- www.yahoo.com internet address = 68.142.226.94
- www.yahoo.com internet address = 68.142.226.98
- www.yahoo.com internet address = 68.142.226.102

P2P file sharing
- Alice runs P2P client application such as Kazaa on her notebook computer.
- Intermittently connects to Internet; gets new IP address for each connection.
- Asks for “Hey Jude”
- Application displays peers that have a copy.
- Alice chooses one of the peers, Bob.
- File is copied from Bob’s PC to Alice’s notebook: HTTP
- While Alice downloads, other users upload other songs from Alice.
- Alice’s application is both a Web client and a transient Web server.
- All peers are servers = highly scalable.
Centralized directory

- Original "Napster" design
- 1) when peer connects, it informs central server:
  - IP address
  - content
- 2) Alice queries server for "Hey Jude"
- 3) Server tells Alice which users (including Bob) have the file
- 4) Alice requests file from Bob

Problems with centralized directory

- Single point of failure
- Performance bottleneck
- Copyright infringement
- file transfer is decentralized, but locating content is highly centralized

Query flooding: Gnutella

- Fully distributed
- No central server
- Public domain protocol
- Many Gnutella clients implementing protocol
- Overlay network is a graph:
  - edge between peer X and Y if there's a TCP connection
  - Edge is not a physical link
- All active peers and edges form overlay net
- Given peer will typically be connected with at least 10 overlay neighbors

Gnutella: protocol

- Query message sent over existing TCP connections
- Peers forward Query message
- QueryHit sent over reverse path
- Scalability: limited scope flooding

Gnutella: Peer joining

1. Joining peer X must find some other peer in Gnutella network: use list of candidate peers
2. X sequentially attempts to make TCP with peers on list until connection setup with Y
3. X sends Ping message to Y; Y forwards Ping message
4. All peers receiving Ping message respond with Pong message
5. X receives many Pong messages. It can then setup additional TCP connections

Exploiting heterogeneity: KaZaA

- Each peer is either a group leader or assigned to a group leader:
  - TCP connection between peer and its group leader
  - TCP connections between some pairs of group leaders
- Group leader tracks the content in all its children.
KaZaA: Querying

- Each file has a hash and a descriptor
- Client sends keyword query to its group leader
- Group leader responds with matches:
  - For each match: metadata, hash, IP address
- If group leader forwards query to other group leaders, they respond with matches
- Client then selects files for downloading
  - HTTP requests using hash as identifier sent to peers holding desired file

KaZaA tricks

- Limitations on simultaneous uploads
- Request queueing
- Incentive priorities
- Parallel downloading

For more info:
J. Liang, R. Kumar, K. Ross, "Understanding KaZaA,”
(available via cis.poly.edu/~ross)