Application-Layer Protocols

The Domain Name System (DNS)

- Computers (hosts, routers) connected to the Internet have two forms of names:
  - IP address — a 32 bit identifier used for addressing hosts and routing data to them
  - Hostname — an ASCII string used by applications
- The DNS is an Internet-wide service that provides mappings between IP addresses and hostnames
  - The DNS is a distributed database implemented in a hierarchy of name servers
  - The DNS is also an application-layer protocol
- Hosts and routers use name servers to resolve names (address/name translation)
  - Name resolution is an essential Internet function implemented as application-layer protocol

The Domain Name System

Services

- Host Aliasing
  - canonical hostname: relay1.west-coast.enterprise.com
- Mail Server Aliasing
  - email address: bob@hotmail.com
  - mail server: relay1.west-coast.hotmail.com
- Load Distribution
  - set of IP addresses associated with 1 canonical hostname (e.g., cnn.com)
  - server response with whole set, but rotates ordering
Names are valuable!

**Sale: Web Address, Unused, Not Cheap**

The Internet address is for sale. The owner of sites like **example.com**, **example.net**, **example.org**, and **example.mil** has an estimated value of $1 million. The estimated value comes from the fact that this domain name has been relatively free for the past 8 years.

**A New York Times article from August 22, 2000**

---

**Name Hierarchy in DNS**

- **hostname** = “dot” separated concatenation of domain names along path toward the root
  - clemson.edu
  - shadow1.cs.clemson.edu
  - cs.clemson.edu

---

**Growth of DNS registrations**

- Source: Internet Software Consortium (http://www.isc.org/)

---

**Top level domains**

- **Generic domains:**
  - .com, .org, .net, .edu, .gov, .mil, .int
  - .biz, .info, .name, .pro
- **Special sponsored names**
  - .aero, .coop, .museum
- **Country code domains**
  - .uk, .de, .jp, .us, etc.
**DNS Overview**

- Applications need IP address to open connection
- Use DNS to find the IP address given a hostname
- Steps:
  1. Application invokes DNS (gethostbyname())
  2. DNS application in host sends query into network (UDP port 53)
  3. DNS application in host receives reply with IP address (after some delay)
  4. IP address passed up to the application

  DNS is a black box as far as the application is concerned.

---

**The Domain Name System**

**Designing a distributed service**

- Why not centralize the DNS
  - A server process on a big, well connected supercomputer?
- Centralized systems do not scale!
  - Poor reliability: centralized = single point of failure
  - Poor performance: centralized = “remote access” for most users
  - Difficult to manage: centralized = all traffic goes to one location, a large staff has to be present to handle registrations
- A centralized system is not politically feasible in an international network

---

**Designing a Distributed Service**

**DNS Name Servers**

- No server has every hostname-to-IP address mapping
- Authoritative name server:
  - Every host is registered with at least one authoritative server that stores that host’s IP address and name
  - The authoritative name server can perform name/address translation for that host’s name/address
- **Local authoritative name servers:**
  - Each ISP, university, company, has a local (default) name server authoritative for its own hosts
  - Resolvers always query a name server local to it to resolve any host name

**Root name servers**

- A root name server is contacted when a local name server that can’t resolve a name
  - The root server either resolves the name or provides pointers to authoritative servers at lower level of name hierarchy
- There are 13 root name servers worldwide
DNS Name Servers
Generic TLD servers (Verisign Corp.)

12 independent sites
- .com, .org, .net server locations (separated from root servers)

DNS Name Servers
Recursive vs. Iterative Queries
- The DNS supports two paradigms of queries:
  - Recursive queries
  - Iterative queries
- Recursive queries place the burden of name resolution (recursively) on the contacted server
- In an iterated query the contacted server simply replies with the name of the server to contact
  - “I don’t know: trying asking X”

DNS Name Servers
Using a server hierarchy for resolving names
- Host shadow1.cs.clemson.edu wants to know the IP address of www.yahoo.com
  - shadow1 contacts its local DNS server tornado.cs.clemson.edu
- To resolve a non-local name, the local name server queries the root server
- The root server responds with the TLD for .com
- The local DNS server contacts the TLD server
- The local DNS server contacts the authoritative server dns.yahoo.com
- Results feed back to shadow1
- shadow1 can now use the IP address of www.yahoo.com to make a connection

DNS Name Servers
Recursive vs. Iterative Queries
- Any query can be recursive or iterative
- Iterative and recursive queries can be combined
- Typically, the query from the requesting host to the local DNS server is recursive and the remaining queries are iterative
DNS Name Servers

Caching and updating DNS entries

- Every server caches all the mappings it learns
  » Cache entries are “soft state”
  » They timeout (are deleted) after some time period
- DNS cache update/notify mechanisms under design by the IETF
  » See RFC 2136

DNS Name Servers

DNS resource records

- The DNS is a distributed database storing resource records (RRs)
- Type = A
  » name is a hostname
  » value is hostname’s IP address
- Type = NS
  » name is a domain
  » value is name of authoritative name server for this domain
- Type = CNAME
  » name is an alias name for some “canonical” (the real) name
  » value is canonical name
- Type = MX
  » value is name of mail server host associated with name

DNS Name Servers

DNS resource records

RR format: <name, value, type, time_to_live>

<table>
<thead>
<tr>
<th>Identification</th>
<th>Flags</th>
<th>Questions</th>
<th>Answers</th>
<th>Additional Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of questions</td>
<td>Number of answer RRs</td>
<td>Number of additional RRs</td>
<td>Questions (variable number)</td>
<td>Answers (variable number of records)</td>
</tr>
</tbody>
</table>

DNS Protocol

DNS query and reply messages

- DNS query and reply messages both have the same message format
- Messages have a fixed length message header
  » Identification — 16 bit query/reply identifier used to match replies to queries
  » Flags:
    ♦ Query/Reply bit
    ♦ "Reply is authoritative" bit
    ♦ "Recursion desired" bit
    ♦ ...
**DNS Protocol**

**DNS query and reply messages**

- Messages have a variable-length "question & answer" body
- Questions:
  - The name and type fields (type A or MX) for a query — hotmail.com MX
- Answers:
  - One RR for each IP address answering query
- Authority:
  - Resource records of other authoritative servers

---

**DNS Resource Records**

**nslookup query/reply message example**

shadow1 nslookup
Default Server: tornado.cs.clemson.edu
Address: 130.127.46.3

> set debug
> www.yahoo.com
Server: tornado.cs.unc.edu
Address: 130.127.46.3

; res_mkquery(QERD, www.yahoo.com, IN, A)
---------
Set answer:
MESSAGE:
opcode = QUERY, id = 16795, rcode = NOERROR
header flags: recursion wanted, recursion available
questions = 1, answers = 9, authority records = 11, additional = 3
QUESTIONS:
  www.yahoo.com, type = A, class = IN

ANSWERS:
- www.yahoo.com
canonical name = www.yahoo.akadns.net
ttl = 300 (s)

- www.yahoo.akadns.net
internet address = 216.109.118.72
ttl = 60 (s)
- www.yahoo.akadns.net
internet address = 216.109.117.106
ttl = 60 (s)
- www.yahoo.akadns.net
internet address = 216.109.117.107
ttl = 60 (s)
- www.yahoo.akadns.net
internet address = 216.109.117.109
ttl = 60 (s)
- www.yahoo.akadns.net
internet address = 216.109.118.64
ttl = 60 (s)
- www.yahoo.akadns.net
internet address = 216.109.118.65
ttl = 60 (s)
- www.yahoo.akadns.net
internet address = 216.109.118.67
ttl = 60 (s)

AUTHORITY RECORDS:
- skadns.net
  nameserver = sz.skadns.org
  ttl = 113598 (s)
- skadns.net
  nameserver = sf.skadns.org
  ttl = 113598 (s)
- skadns.net
  nameserver = sh.skadns.org
  ttl = 113598 (s)
- skadns.net
  nameserver = eu3.skam.net
  ttl = 113598 (s)
- skadns.net
  nameserver = use2.skam.net
  ttl = 113598 (s)
- skadns.net
  nameserver = use3.skam.net
  ttl = 113598 (s)
- skadns.net
  nameserver = use5.skam.net
  ttl = 113598 (s)

ADDITIONAL RECORDS:
- za.skadns.org
  internet address = 208.185.132.176
ttl = 126017 (s)
- use4.skam.net
  internet address = 80.67.67.182
ttl = 126017 (s)
- usew.skam.net
  internet address = 63.241.73.214
ttl = 126017 (s)

Non-authoritative answer:
Name: www.yahoo.akadns.net
Aliases: www.yahoo.com
DNS Example

DNS processing for an iterated query

- Resolve the hostname in http://www.cnn.com

Local Server

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.</td>
<td>NS</td>
<td>A.ROOT-SERVERS.NET</td>
</tr>
<tr>
<td>cnn.com</td>
<td>NS</td>
<td>DNS.CNN.COM</td>
</tr>
<tr>
<td><a href="http://www.cnn.com">www.cnn.com</a></td>
<td>A</td>
<td>207.25.71.28</td>
</tr>
</tbody>
</table>

Local Server

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>IP Address</th>
</tr>
</thead>
</table>

The Domain Name System

Summary

- F gets 270,000,000+ hits per day
  - Other servers have comparable load
- The Verisign TLD servers answer 5,000,000,000 queries per day
- Clearly the DNS would collapse without:
  - Hierarchy
  - Distributed processing
  - Caching
- If DNS fails, Internet services stop working!

For More Info

- DNS Specification
  - RFC 1034, RFC 1035
  - [Mockapetris 1988]
- DNS Caching
  - RFC 2136
- DNS Attacks
  - CAIDA, Nameserver DoS Attack October 2 analysis
  - www.caida.org/projects/dns-analysis/oct02dos.xml
**Application-Layer Protocols**

**Peer-to-Peer (P2P) Architecture**

- No always-on server
- Arbitrary end systems directly communicate
- Peers are intermittently connected and change IP addresses

**Application-Layer Protocols**

**P2P File Sharing**

- P2P file sharing accounts for more traffic than any other application on the Internet
- Types of files shared
  - MP3s 3-8 MB
  - videos 10MB - 1GB
  - software
- Legality
  - [von Lohmann 2003]

**Application-Layer Protocols**

**P2P File Sharing / Example**

- Alice runs P2P client application on her notebook computer
- Intermittently connects to Internet; gets new IP address
- Asks for “Hey Jude”
- Application displays other peers that have copy of Hey Jude.
- Alice chooses one of the peers, Bob.
- File is copied from Bob’s PC to Alice’s notebook: HTTP
- While Alice downloads, other users uploading from Alice.
- Alice’s peer is both a Web client and a transient Web server.

**Application-Layer Protocols**

**P2P File Sharing**

**Content Location Problem**

- How do peers know which other peers have content that they’re interested in?
- Centralized Directory
- Query Flooding
- Heterogeneous Approach
## P2P File Sharing

### Centralized Directory (Napster)

1. When peer connects, it informs central server:
   - IP address
   - content
2. Alice queries for “Hey Jude”
3. 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.

## P2P File Sharing

### Query Flooding (Gnutella)

- Fully distributed
  - no central server
- Public domain protocol
- Many Gnutella clients implementing protocol
- Peers form an overlay network
  - edge between peer X and Y if there’s a TCP connection
  - group of active peers and edges is overlay network
  - edge is not a physical link
  - given peer will typically be connected with < 10 overlay neighbors

Protocol specification: [Gnutella 2004]

### Query Flooding Messages

- **Query** message sent over existing TCP connections
- Peers forward **Query** message
- **QueryHit** message sent over reverse path

Scalability: limited scope flooding

File transfer: HTTP
P2P File Sharing

KaZaA

- Borrows ideas from Napster and Gnutella
  - no dedicated server
  - peers are either normal peers or group leaders
  - each group leader is a mini-Napster-like hub
  - group leaders create a network that resembles a Gnutella network
- Contributes more traffic to the Internet than any other application
  - [Gummadi 2003, Saroiu 2002]
- Proprietary specification
  - harder to know exactly how it works
  - [Liang 2004] - KaZaA traffic measurement
- Encrypts control traffic

P2P File Sharing

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: 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 queuing
- Incentive priorities
  - priority in queue given to users who upload more than download
- Parallel downloading
  - use byte-range field in HTTP request header
**Peer-to-Peer Systems**

*For More Info*

- Content Location
  - [Stoica 2001]
  - [Rowstron 2001]
  - [Ratnasamy 2001]
  - [Zhao 2004]
  - [Maymounkov 2002]
  - [Garces-Erce 2003]

- Gnutella
  - [Gnutella 2004] - protocol specification

- KaZaA
  - [KaZaA 2004]
  - [Gummadi 2003]
  - [Saroiu 2002]
  - [Liang 2004]

**Application-Layer Protocols**

*Chapter 2 Summary*

- Socket Programming
- Client-Server Architecture
  - HTTP (caching, generating synthetic traffic)
  - FTP
  - SMTP
  - POP3
  - DNS

- Peer-to-Peer Systems
  - Napster (centralized directory)
  - Gnutella (query flooding)
  - KaZaA (combined approach)