Applications & Application-Layer Protocols: Email (SMTP) and DNS

Dr. Michele Weigle
Department of Computer Science
Old Dominion University
mweigle@cs.odu.edu

http://www.cs.odu.edu/~mweigle/CS312-F11

Application-Layer Protocols Outline

- The architecture of distributed systems
  - Client/Server computing
  - P2P and Hybrid computing
- Example client/server systems and their application-layer protocols
  - The World-Wide Web (HTTP)
  - E-mail (SMTP & POP)
  - Internet Domain Name System (DNS)
Application-Layer Protocols

Electronic mail

- Major components:
  - User agents
  - Mail servers
  - Mailboxes

- Protocols:
  - Simple Mail Transfer Protocol (SMTP) delivers mail to servers
    - From clients to local mail server
    - Inter-mail server delivery
  - Post Office Protocol (POP) for user access to delivered email

Electronic Mail

Mail servers

- Servers maintain:
  - A message queue of outgoing email messages
  - A mailbox containing incoming messages for each user

- SMTP protocol is run between mail agents and servers to send email messages
  - Client — the sending mail server or agent
  - Server — the receiving mail server
Electronic Mail

The email delivery process

- User’s mail agent contacts its local mail server
- Local mail server contacts the destination mail server(s)
- Destination mail server places the mail into the appropriate user’s mailbox
- User retrieves mail via a mail access protocol

The Email Delivery Process

SMTP [RFC 2821]

- SMTP uses a TCP socket on port 25 to transfer email reliably from client to server
- Email is temporarily stored on the local server and eventually transferred directly to receiving server
  » Intermediate relay is a special case
- Three phases of the protocol:
  » Handshaking (“greeting”)
  » Transfer of messages
  » Closure
- Client/server interaction follows a command/response paradigm
  » Commands are plain ASCII text
  » Responses are a status code and an optional phrase
  » Command and response lines terminated with CRLF
The Email Delivery Process
Sample SMTP interaction

- SMTP client establishes TCP connection to server hamburger.edu at port 25
  » (SMTP is non-standard in that the server “talks first”)

Server: 220 hamburger.edu
Client: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection

Electronic Mail
Mail message format (RFC 2822)

- Header lines, e.g.,
  » From:
  » To:
  » Subject:
    these are different from SMTP commands!

- Body
  » The “message”, ASCII characters only
Electronic Mail

Mail message format example

Server: 220 hamburger.edu
Client: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: From: alice@crepes.fr
C: To: bob@hamburger.edu
C: Subject: food
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection

Application-Layer Protocols

HTTP v. SMTP

◆ HTTP is a “pull” protocol, SMTP is a “push” protocol

◆ Persistence:
   » SMTP uses persistent connections
   » HTTP may or may not

◆ Message/object content:
   » Both have ASCII command/response interaction and status codes
   » SMTP requires that messages be in 7-bit ASCII
   » HTTP can transfer anything — each object is encapsulated in its own response headers
Electronic Mail

Mail access protocols

- SMTP: Delivery to receiver’s server
- Mail access protocol: Retrieval from server by a user
  - POP [RFC 1939] — Authorization and download
  - IMAP (Internet Mail Access Protocol) [RFC 2060]
    - More features (more complex)
    - Manipulation of stored messages on server
  - HTTP: Gmail, Yahoo! Mail, etc.

Mail Access Protocols

POP-3 vs. IMAP

**POP3**

- Downloads messages to the client
- “Download-and-delete”: User cannot re-read e-mail if he changes client
- “Download-and-keep”: copies of messages on different clients
- POP3 is stateless across sessions

**IMAP**

- Keep all messages in one place: the server
- Allows user to organize messages in folders
- IMAP keeps user state across sessions:
  - names of folders and mappings between message IDs and folder name
Mail Access Protocols
Web-Based Email

◆ User agent is a web browser
◆ User communicates with remote mailbox via HTTP
◆ When recipient wants to access a message
  » email message is sent from mail server to browser using HTTP
◆ When sender wants to send a message
  » email is sent from browser to mail server using HTTP
  » mail server still uses SMTP to communicate with other mail servers

Application-Layer Protocols
Outline

◆ The architecture of distributed systems
  » Client/Server computing
  » P2P and Hybrid computing
◆ Example client/server systems and their application-layer protocols
  » The World-Wide Web (HTTP)
  » E-mail (SMTP & POP)
  » Internet Domain Name System (DNS)
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 an application-layer protocol

The Domain Name System
Services

◆ **Host Aliasing**
  » *canonical hostname*: relay1.west-coast.enterprise.com
  » *aliases*: enterprise.com, www.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
The Domain Name System

Name Hierarchy in DNS

hostname = “dot” separated concatenation of domain names along path toward the root

- odu.edu
- antares.cs.odu.edu
- cs.odu.edu

Name Hierarchy in the DNS

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() in C)
   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

DNS Name Servers

Root name servers

- A root name server is contacted when a local name server 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.)

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

DNS Name Servers
Using a server hierarchy for resolving names

- Host deneb.cs.odu.edu wants to know the IP address of www.yahoo.com
  > deneb contacts its local DNS server wrath.cs.odu.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 deneb
- deneb can now use the IP address of www.yahoo.com to make a connection
DNS Name Servers
DNS resource records

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

- 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 / Examples

- A record
  » (relay1.west-coast.yahoo.com, 145.137.93.126, A)

- NS record
  » (yahoo.com, dns.yahoo.com, NS)

- CNAME record
  » (yahoo.com, relay1.west-coast.yahoo.com, CNAME)

- MX record
  » (yahoo.com, mail.yahoo.com, MX)
**DNS**

**Inserting Records into DNS**

- Example: new startup “Network Utopia”
  - Register name networkutopia.com at *DNS registrar* (e.g., Network Solutions)
    - provide name, IP address of authoritative name server
    - registrar inserts two RRs into com TLD server:
      - (networkutopia.com, dns.networkutopia.com, NS)
      - (dns.networkutopia.com, 212.212.212.1, A)

- On dns.networkutopia.com, create records for web and email
  - (www.networkutopia.com, 212.212.212.2, A)
  - (networkutopia.com, www.networkutopia.com, CNAME)
  - (networkutopia.com, mail.networkutopia.com, MX)
  - (mail.networkutopia.com, 212.212.212.3, A)

- *How do people get IP address of your Web site?*

**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 servers cache entries, end-systems do not (though individual applications might)
**DNS Example**

**DNS processing**

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

```
<table>
<thead>
<tr>
<th>com.</th>
<th>NS</th>
<th>A.GTLD-SERVERS.NET</th>
<th>192.5.6.30</th>
</tr>
</thead>
<tbody>
<tr>
<td>cnn.com</td>
<td>NS</td>
<td>DNS.CNN.COM</td>
<td>128.2.232.1</td>
</tr>
<tr>
<td><a href="http://www.cnn.com">www.cnn.com</a></td>
<td>A</td>
<td>207.25.71.28</td>
<td></td>
</tr>
</tbody>
</table>
```

- Resolve the hostname `www2.cnn.com` with a warm cache

```
<table>
<thead>
<tr>
<th>com.</th>
<th>NS</th>
<th>A.GTLD-SERVERS.NET</th>
<th>192.5.6.30</th>
</tr>
</thead>
<tbody>
<tr>
<td>cnn.com</td>
<td>NS</td>
<td>DNS.CNN.COM</td>
<td>128.2.232.1</td>
</tr>
</tbody>
</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!

Application-Layer Protocols

Outline

- The architecture of distributed systems
  - Client/Server computing
  - P2P computing
  - Hybrid (Client/Server and P2P) systems
- Example client/server systems and their application-level protocols
  - The World-Wide Web (HTTP)
  - E-mail (SMTP & POP)
  - Internet Domain Name System (DNS)