Chapter 8
The Application Layer

8.0 Applications and Application-layer Protocols
8.1 DSN-The Domain Name System
8.0 Applications and Application-layer Protocols

Application: communicating, distributed processes
- running in network hosts in "user space"
- exchange messages to implement applications
- e.g., email, file transfer, the Web

Application-layer protocols
- one "piece" of an app
- define messages exchanged by applications and actions taken
- user services provided by lower layer protocols

An application-layer protocol defines how an application’s processes, running on different end systems, and passing messages to each other. In particular, an application-layer protocol defines:

A. The types of messages exchanged, e.g. request messages and response messages.
B. The syntax of the various message types, such as the fields in the message and how the fields are delineated.
C. The semantics of the fields, i.e. the meaning of the information in the fields.
D. Rules for determining when and how a process sends messages and responds to messages.
Client and Server Sides of an Application

The host that initiates the session is labeled the client.

Fig 8.2 Client-server information
Network applications: some jargon

- A **process** is a program that is running within a host.
- Within the same host, two processes communicate with **interprocess communication** defined by the OS.
- Processes running in different hosts communicate with an **application-layer protocol**

- A **user agent** is an interface between the user and the network application.
  - Web: browser
  - E-mail: mail reader
  - Streaming audio/video: media player
API: application programming interface

- defines interface between application and transport layers
- socket: Internet API

  - two processes communicate by sending data into socket, reading data out of socket

Q: how does a process “identify” the other process with which it wants to communicate?

  - IP address of host running other process
  - “port number” - allows receiving host to determine to which local process the message should be delivered
Processes Communicating Across a Network

A process sends messages into, and receives messages from the network through its socket. A process’s socket can be thought of as the process’s door.

The sending process shoves the message out its door and through a transportation infrastructure to the receiving process’s door (socket).
Fig 8.3 Application processes, sockets, and underlying transport protocol
The services a network application needs from a transport protocol

A. reliable data transfer
electronic mail, file transfer, remote host access, financial application…

B. Bandwidth
bandwidth-sensitive applications
such as multimedia
elastic applications
such as electronic mail, file transfer…

C. Timing
such as Internet telephony, multiplayer games
Transport service requirements of common applications

<table>
<thead>
<tr>
<th>Application</th>
<th>Data loss</th>
<th>Bandwidth</th>
<th>Time Sensitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>file transfer</td>
<td>no loss</td>
<td>elastic</td>
<td>no</td>
</tr>
<tr>
<td>e-mail</td>
<td>no loss</td>
<td>elastic</td>
<td>no</td>
</tr>
<tr>
<td>Web documents</td>
<td>loss-tolerant</td>
<td>elastic</td>
<td>no</td>
</tr>
<tr>
<td>real-time audio/video</td>
<td>loss-tolerant</td>
<td>audio: 5Kb-1Mb</td>
<td>yes, 100’s msec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>video:10Kb-5Mb</td>
<td></td>
</tr>
<tr>
<td>stored audio/video</td>
<td>loss-tolerant</td>
<td>same as above</td>
<td>yes, few secs</td>
</tr>
<tr>
<td>interactive games</td>
<td>loss-tolerant</td>
<td>few Kbps up</td>
<td>yes, 100’s msec</td>
</tr>
<tr>
<td>financial apps</td>
<td>no loss</td>
<td>elastic</td>
<td>yes and no</td>
</tr>
</tbody>
</table>
## Internet applications: their protocols and transport protocols

<table>
<thead>
<tr>
<th>Application</th>
<th>Application layer protocol</th>
<th>Underlying transport protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>e-mail</td>
<td>SMTP [RFC 821]</td>
<td>TCP</td>
</tr>
<tr>
<td>remote terminal access</td>
<td>Telnet [RFC 854]</td>
<td>TCP</td>
</tr>
<tr>
<td>Web</td>
<td>HTTP [RFC 2068]</td>
<td>TCP</td>
</tr>
<tr>
<td>file transfer</td>
<td>FTP [RFC 959]</td>
<td>TCP</td>
</tr>
<tr>
<td>streaming multimedia</td>
<td>proprietary</td>
<td>TCP or UDP</td>
</tr>
<tr>
<td></td>
<td>(e.g. RealNetworks)</td>
<td></td>
</tr>
<tr>
<td>remote file server</td>
<td>NFS</td>
<td>TCP or UDP</td>
</tr>
<tr>
<td>Internet telephony</td>
<td>proprietary</td>
<td>typically UDP</td>
</tr>
<tr>
<td></td>
<td>(e.g., Vocaltec)</td>
<td></td>
</tr>
</tbody>
</table>
SMTP : Simple Mail Transfer Protocol
HTTP : Hypertext Transfer Protocol
FTP : File Transfer Protocol
NFS : Network File System
Telnet : A character – oriented protocol that uses a standard encoding when it transfers data.
8.1 DNS: Domain Name System
(The Internet’s Directory Service)

People: many identifiers:
  – Social Security Number, name, Passport #

Internet hosts, routers:
  – IP address (32 bit) - used for addressing datagrams
  – “name”, e.g., gaia.cs.umass.edu
    - used by humans

Q: map between IP addresses and name?

Domain Name System:

a) distributed database implemented in hierarchy of many name servers

b) application-layer protocol host, routers, name servers to communicate to resolve names (address/name translation)
  – note: core Internet function implemented as application-layer protocol
  – complexity at network's “edge”
8.1.1 The Domain Name Space

A. The Internet is divided into over 200 top-level domains, where each domain covers many hosts.

B. Each domain is partitioned into subdomains, and these are further partitioned, and so on.

C. All these domains can be represented by a tree Fig 8.4

D. The top-level domain comes in two flavors: generic and countries. But in November 2000 biz, info, name (people’s names), and Pro (professions) are added.

E. The requester of a second-level domain name pays a small annual fee.

F. Virtually, every common (English) word has been taken in the com domain.

G. Each domain is named by the path upward from it to the root (unnamed)

H. Domain names are case insensitive (edu, Edu, EDU)
The DNS Name Space

A leaf domain may contain a simple host, or it may represent a company and contain thousand hosts.

Fig 8.4 A portion of the Internet domain name space.
DNS name servers

Why not centralize DNS?

a) single point of failure
b) traffic volume
c) distant centralized database
d) maintenance

doesn't scale!

no server has all name-to-IP address mappings

local name servers:

- each ISP, company has *local (default) name server*
- host DNS query first goes to local name server

authoritative name server:

- for a host: stores that host's IP address, name
- can perform name/address translation for that host's name
8.1.2 Resource Records

A. Every domain (single host or a top-level domain) can have a set of resource records associated with it.

B. A source record is a five-tuple which is represented as ASCII text

Domain-name  Time-to-live  Class  Type  Value

Class: For Internet, it is always IN,

For Non-Internet information, other codes can be used (rarely seen)
Resource Records (RR)

The Type field tells what kind of record this is

<table>
<thead>
<tr>
<th>Type</th>
<th>Meaning</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOA</td>
<td>Start of Authority</td>
<td>Parameters for this zone</td>
</tr>
<tr>
<td>A</td>
<td>IP address of a host</td>
<td>32-Bit integer</td>
</tr>
<tr>
<td>MX</td>
<td>Mail exchange</td>
<td>Priority, domain willing to accept e-mail</td>
</tr>
<tr>
<td>NS</td>
<td>Name Server</td>
<td>Name of a server for this domain</td>
</tr>
<tr>
<td>CNAME</td>
<td>Canonical name</td>
<td>Domain name</td>
</tr>
<tr>
<td>PTR</td>
<td>Pointer</td>
<td>Alias for an IP address</td>
</tr>
<tr>
<td>HINFO</td>
<td>Host description</td>
<td>CPU and OS in ASCII</td>
</tr>
<tr>
<td>TXT</td>
<td>Text</td>
<td>Uninterpreted ASCII text</td>
</tr>
</tbody>
</table>

Fig 8.5 The principal DNS resource records types.
SOA: the name server’s zone, e-mail address of its administrator, a unique serial number, and various flags and timeouts

A: a 32-bit IP address for some host
   some hosts may have multiple addresses

MX: The name of the host prepared to accept e-mail for the specified domain

NS: specifying name servers of this domain

CNAME: allowing aliases to be created
   e.g. paul@cs.mit.edu←alias
       paul@lcs.mit.edu
       cs.mit.edu 8400 IN CNAME lcs.mit.edu

PTR: providing reverse lookups (IP→e-mail addr.)

HINFO: what kind of machine and operating system a domain corresponds to

TXT: identifying the domain in arbitrary ways
DNS records

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

- **Type=A**
  - *name* is hostname
  - *value* is IP address

- **Type=NS**
  - *name* is domain (e.g. foo.com)
  - *value* is IP address of authoritative name server for this domain

- **Type=CNAME**
  - *name* is an alias name for some “cannonical” (the real) name
  - *value* is cannonical name

- **Type=MX**
  - *value* is hostname of mailserver associated with *name*
<table>
<thead>
<tr>
<th>Host</th>
<th>Type</th>
<th>Class</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>cs.vu.nl.</td>
<td>SOA</td>
<td>86400</td>
<td>star boss (952771,7200,7200,2419200,86400)</td>
</tr>
<tr>
<td>cs.vu.nl.</td>
<td>TXT</td>
<td>86400</td>
<td>&quot;Divisie Wiskunde en Informatica.&quot;</td>
</tr>
<tr>
<td>cs.vu.nl.</td>
<td>TXT</td>
<td>86400</td>
<td>&quot;Vrije Universiteit Amsterdam.&quot;</td>
</tr>
<tr>
<td>cs.vu.nl.</td>
<td>MX</td>
<td>86400</td>
<td>1 zephyr.cs.vu.nl.</td>
</tr>
<tr>
<td>cs.vu.nl.</td>
<td>MX</td>
<td>86400</td>
<td>2 top.cs.vu.nl.</td>
</tr>
<tr>
<td>flits.cs.vu.nl.</td>
<td>HINFO</td>
<td>86400</td>
<td>Sun Unix</td>
</tr>
<tr>
<td>flits.cs.vu.nl.</td>
<td>A</td>
<td>86400</td>
<td>130.37.16.112</td>
</tr>
<tr>
<td>flits.cs.vu.nl.</td>
<td>A</td>
<td>86400</td>
<td>192.31.231.165</td>
</tr>
<tr>
<td>flits.cs.vu.nl.</td>
<td>MX</td>
<td>86400</td>
<td>1 flits.cs.vu.nl.</td>
</tr>
<tr>
<td>flits.cs.vu.nl.</td>
<td>MX</td>
<td>86400</td>
<td>2 zephyr.cs.vu.nl.</td>
</tr>
<tr>
<td>flits.cs.vu.nl.</td>
<td>MX</td>
<td>86400</td>
<td>3 top.cs.vu.nl.</td>
</tr>
<tr>
<td><a href="http://www.cs.vu.nl">www.cs.vu.nl</a>.</td>
<td>CNAME</td>
<td>86400</td>
<td>star.cs.vu.nl</td>
</tr>
<tr>
<td>ftp.cs.vu.nl.</td>
<td>CNAME</td>
<td>86400</td>
<td>zephyr.cs.vu.nl</td>
</tr>
<tr>
<td>rowboat</td>
<td>A</td>
<td></td>
<td>130.37.56.201</td>
</tr>
<tr>
<td></td>
<td>MX</td>
<td></td>
<td>1 rowboat</td>
</tr>
<tr>
<td></td>
<td>MX</td>
<td></td>
<td>2 zephyr</td>
</tr>
<tr>
<td></td>
<td>HINFO</td>
<td></td>
<td>Sun Unix</td>
</tr>
<tr>
<td>little-sister</td>
<td>A</td>
<td></td>
<td>130.37.62.23</td>
</tr>
<tr>
<td></td>
<td>HINFO</td>
<td></td>
<td>Mac MacOS</td>
</tr>
<tr>
<td>laserjet</td>
<td>A</td>
<td></td>
<td>192.31.231.216</td>
</tr>
<tr>
<td></td>
<td>HINFO</td>
<td></td>
<td>&quot;HP Laserjet III Si&quot; Proprietary</td>
</tr>
</tbody>
</table>

A portion of a possible DNS database for \textit{cs.vu.nl}.
8.1.3 Name Servers

The DNS name space is divided into non-overlapping zones. Each zone contains some part of the tree and name servers holding the information about that zone. A zone has one primary name server and more secondary name servers, which get their information from the primary name server.

Fig 8.6 Part of the DNS name space showing the division into zones.
Root name servers

a) contacted by local name server that can not resolve name

b) root name server:
   - contacts authoritative name server if name mapping not known
   - gets mapping
   - returns mapping to local name server

c) ~ dozen root name servers worldwide
8.1.4 DNS Operation

- A user program requests an IP address for a domain name.
- A resolver module in the local host or local ISP formulates a query for a local name server in the same domain as the resolver.
- The local name server checks to see if the name is in its local database or cache, and, if so, returns the IP address to the requestor. Otherwise, the name server queries other available name servers, starting down from the root of the DNS tree or as high up the tree as possible.
- When a response is received at the local name server, it stores the name/address mapping in its local cache and may maintain this entry for the amount of time specified in the time to live field of the retrieved RR.
- The user program is given the IP address or an error message.
DNS example

Root name server:

a) may not know authoritative name server
b) may know intermediate name server: who to contact to find authoritative name server
Fig 8.7 A query chain with recursive queries and iterative queries.