Application-layer Protocols

Kai Shen
Dept. of Computer Science, University of Rochester

Network Applications and Application-layer Protocols

Network applications:
- running in end systems (hosts)
- distributed, communicating using network
- e.g., e-mail, Web, FTP, instant messaging

Application-layer protocols
- define messages exchanged by apps and actions taken
- use communication services provided by lower layer protocols (TCP, UDP)

Protocols for the Internet Architecture

<table>
<thead>
<tr>
<th>application</th>
<th>HTTP</th>
<th>SMTP</th>
<th>DNS</th>
<th>Joe’s protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>transport</td>
<td>TCP</td>
<td>UDP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>network</td>
<td>IP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>link</td>
<td>Ethernet</td>
<td>802.11</td>
<td>...</td>
<td>PPP</td>
</tr>
<tr>
<td>physical</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Protocols vs. Implementations

- They specify:
  - Syntax and semantics of messages exchanged
  - Rules for when and how processes send & respond to messages

- They don’t specify the implementation:
  - Programming, data structure, ...

- The goal:
  - Components implemented independently can inter-operate with each other as long as they follow the protocol specification
Network Applications

- Traditional applications
  - Web and HTTP
  - FTP
  - Electronic mail: SMTP, POP3, IMAP
  - DNS

- New applications
  - Internet load balancing
  - Web caching
  - Content-distribution networks
  - Peer-to-peer and overlay networks

HTTP and Web

- HTTP: hypertext transfer protocol
- Web: the application using HTTP protocol
- Client/server model
  - client: browser that requests, receives, "displays" Web objects
  - server: Web server sends objects in response to requests
- HTTP 1.0, HTTP 1.1

HTTP and Web (more)

- HTTP Specifies
  - Types of messages exchanged, e.g., request & response messages
  - Syntax of message types: what fields in messages & how fields are delineated
  - Semantics of the fields, i.e., meaning of information
  - Rules for when and how processes send & respond to messages

- Independently developed Web client and server can interoperate as long as they observe HTTP
  - Web client (browser): IE, Netscape, Mozilla, ...
  - Web server: Apache, Microsoft IIS, Netscape Enterprise, and the one you wrote in assignment #1

HTTP in Action

- Uses TCP:
  - Client initiates TCP connection (creates socket) to server, port 80 default
  - HTTP messages exchanged between browser and Web server using TCP

- Non-persistent HTTP
  - At most one object is sent over a TCP connection.
  - HTTP/1.0 uses non-persistent HTTP

- Persistent HTTP
  - Multiple objects can be sent over single TCP connection between client and server.
  - HTTP/1.1 allows persistent connections
Performance of HTTP Persitency

TCP performance overhead:
- connection establishment
- congestion control: slow start

Non-persistent HTTP issues:
- pay these overhead for each object
- but browsers can open parallel TCP connections to fetch referenced objects

Persistent HTTP
- HTTP messages between same client/server are sent over one TCP connection

Electronic Mail

Two types of components:
- user agents
- mail servers

Two types of protocols:
- mail transfer protocol
  - from sender agent to the receiver's mail server
  - SMTP (simple mail transfer protocol)
- mail access protocol
  - the receiver pulls mails from server to agent
  - POP3, IMAP, HTTP

Mail Servers and User Agents

Mail servers
- mailbox contains incoming messages for users
- message queue of outgoing (to be sent) mail messages

User agents
- a.k.a. "mail reader"
- composing, editing, reading mail messages
- e.g., pine, Outlook, Netscape Messenger

Alice sends message to Bob

1) Alice uses UA to compose message and "to" bob@hotmail.com
2) Alice's UA sends message to her mail server through SMTP; message placed in message queue
3) Alice's mail server opens a SMTP session with Bob's mail server
4) Alice's message is sent to Bob's mail server in the SMTP session
5) Bob's mail server places the message in Bob's mailbox
6) Bob invokes his user agent to read message
SMTP: Simple Mail Transfer Protocol

- SMTP protocol: forward email messages between mail servers
  - "client": sending mail server
  - "server": receiving mail server
- uses TCP to transfer email message from client to server, default port 25
- three phases of transfer:
  - handshaking (greeting)
  - transfer of messages
  - closure
- command/response interaction
  - commands: ASCII text
  - response: status code and phrase
- messages must be in ASCII

SMTP Interaction between mail servers at cs.rochester.edu (sender) and hotmail.com (receiver)

S: 220 hotmail.com
C: HELO cs.rochester.edu
S: 250 Hello cs.rochester.edu, pleased to meet you
C: MAIL FROM: <alice@cs.rochester.edu>
S: 250 alice@cs.rochester.edu... Sender ok
C: RCPT TO: <bob@hotmail.com>
S: 250 bob@hotmail.com ... 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 hotmail.com closing connection

Try SMTP interaction for yourself

telnet servername 25
see 220 reply from server
enter HELO, MAIL FROM, RCPT TO, DATA, QUIT commands

This lets you
- send email without using email client
- ...

Mail Access Protocols

- SMTP: delivery/storage to receiver's server
- Mail access protocol: retrieval from server
  - POP: Post Office Protocol
    - authorization and download
  - IMAP: Internet Mail Access Protocol
    - more features (more complex)
    - manipulation of stored msgs on server
  - HTTP: Hotmail, Yahoo! Mail, etc.
DNS: Domain Name System

People:
- SSN – used for tax reporting, good for bookkeeping, indexing
- Name – human friendly, easy to remember

Internet hosts:
- IP address – used for addressing, routing on the Internet
- "name", e.g., tallinn.cs.rochester.edu - human friendly

Q: map between IP addresses and name?
- DNS query: find the IP address for a given name
- Not part of Internet as a network
- Core function for Internet applications
  - "telnet tallinn.cs.rochester.edu" vs. "telnet 128.151.67.155"

Domain Name System: Overview

- Distributed database
  - Implemented with collaboration of many name servers
  - Distributed all over the network
- DNS queries
  - Name lookup: find the IP address for a given name
  - Others: find mail server for a domain, ...
- Application-layer protocol
  - Implemented on top of transport-layer protocol
  - Complexity at network's "edge": does not involve routers

DNS: Decentralized and Hierarchical

- Decentralized
  - No server has all name-to-IP address mappings
- Why not centralize DNS?
  - Single point of failure, performance bottleneck
  - Can only be close to few hosts → long latency to many others
- What if massively replicating it?
  - Too much data to fit into any single server
  - Hard to make any changes on the data
- Hierarchical
  - There are different types of name servers

Types of Name Servers

- Local name server (defined for a particular group of hosts):
  - Each organization often has local (default) name server
  - Used when one of the hosts under its jurisdiction initiates a DNS query
  - First stop for a DNS query
- Authoritative name server (defined for a particular group of hosts):
  - Maintain "IP address - name" mapping for a group of hosts
  - So it can answer DNS query for these hosts
  - Used when one of the hosts under its jurisdiction is the target of a DNS query
  - Last stop for a DNS query (if ever reached)
  - Often the same as local name server for an organization
- Each local NS may not know all authoritative NSs
  - Something is missing...

10/27/2004
CSC 257/457 - Fall 2004
Root Name Servers

- Root name server (defined globally)
  - bridging between local NSs and authoritative NSs
  - contacted by local name server that can not resolve name
- what it does:
  - contacts authoritative name server
  - gets result and returns it to the requesting local name server

13 root name servers worldwide

Intermediate Name Servers

- Root name server:
  - may not know authoritative name server
  - may know intermediate name server: who to contact to find authoritative name server

DNS Query in Action

1. Contacts its local DNS server, dns.kai.com
2. dns.kai.com contacts root name server
3. root name server contacts authoritative name server, dns.cs.rochester.edu

Note that the destination host is not involved in the lookup.

What is the real difference here?

Iterative Queries

- Recursive query:
  - puts burden of name resolution on contacted name server
- Iterative query:
  - contacted server replies with name of server to contact
  - "I don't know this name, but ask this server"
DNS Caching

- Once (any) name server learns mapping, it caches mapping
  - Next query for the same name can be answered directly
  - Can be at any step of the name lookup (even an end host)
- Cache entries timeout (disappear) after some time
  - Usually two days, can be set differently
  - The reason is that the mapping can change
    - Switch Web server to a more powerful machine ⇒ change the mapping between www.cs.rochester.edu to IP address of the new machine
    - Load balancing
  - Why doesn’t the host notify all name servers that cache its mapping when the mapping changes?

Disclaimer

- Parts of the lecture slides contain original work of James Kurose, Larry Peterson, and Keith Ross. The slides are intended for the sole purpose of instruction of computer networks at the University of Rochester. All copyrighted materials belong to their original owner(s).