

Application-layer Protocols

Kai Shen

10/28/2009 CSC 257/457 - Fall 2009 1

Network Applications and Application-Layer Protocols

Network applications:

- running in end systems (hosts)
- distributed, communicating using network
- use communication services provided by lower layer protocols (TCP, UDP)

Application-layer comm. protocols:

- define comm. interface between application components
- open protocol** ⇒ components implemented independently can inter-operate

10/28/2009 CSC 257/457 - Fall 2009 2

Web and HTTP

- Web:** the application for distributed publishing and viewing of content
- Client/server model
 - server:** hosts published content and sends the content upon request
 - client:** requests, receives, and displays content
- HTTP:** the comm. protocol supporting the web

10/28/2009 CSC 257/457 - Fall 2009 3

Performance of HTTP Persistency

Non-persistent HTTP (1.0)

- At most one object is sent over a TCP connection.
- Pays TCP per-connection overhead for each object.

Persistent HTTP (1.1)

- Multiple objects can be sent over single TCP connection between the browser and web server.

TCP per-connection overhead:

- connection establishment
- congestion control: slow start

10/28/2009 CSC 257/457 - Fall 2009 4

Additional Example Applications

- Remote console
 - Components: host machine and console client
 - Comm. protocol: SSH
- Electronic mail
 - Components: mail servers and mail readers
 - Comm. protocol: SMTP, POP3, IMAP
- Domain name system (DNS)
- Network file system (NFS)
- Peer-to-peer file sharing (Gnutella, BitTorrent)
-

10/28/2009

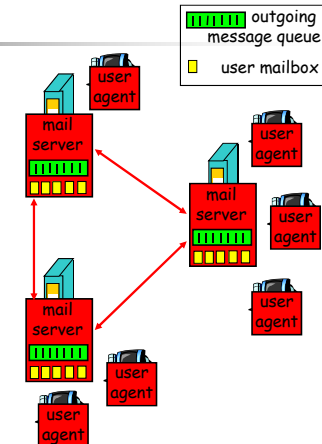
CSC 257/457 - Fall 2009

5

Electronic Mail

Two types of components:

- **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, Web browser



10/28/2009

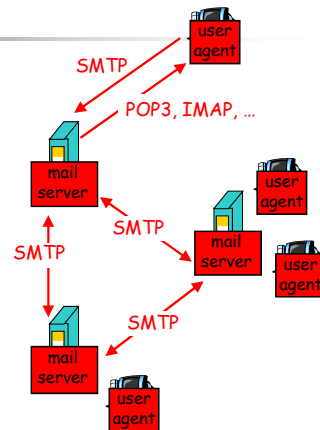
CSC 257/457 - Fall 2009

6

Electronic Mail

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
 - POP, IMAP, proprietary protocol



10/28/2009

CSC 257/457 - Fall 2009

7

SMTP Interaction between mail servers at cs.rochester.edu (client) and gmail.com (server)

```

S: 220 gmail.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@gmail.com>
S: 250 bob@gmail.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 gmail.com closing connection
  
```

10/28/2009

CSC 257/457 - Fall 2009

8

Try SMTP Interaction Yourself

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

This lets you

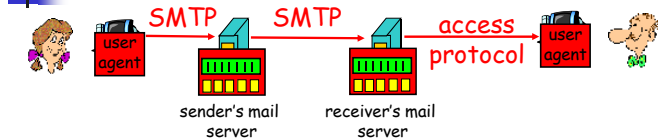
- send email without using a normal email client
- ...

10/28/2009

CSC 257/457 - Fall 2009

9

Mail Access Protocols



- SMTP: delivery/storage to receiver's server
- Mail access protocol: retrieval from server
 - POP3: Post Office Protocol
 - authorization and download
 - IMAP: Internet Mail Access Protocol
 - manipulation of stored messages on server
 - Proprietary: Gmail, etc.

10/28/2009

CSC 257/457 - Fall 2009

10

DNS: Domain Name System

People: multiple identifiers

- SSN - good for machine bookkeeping, indexing
- name - human friendly, easy to remember

Internet hosts:

- IP address - used for addressing, routing on the Internet
- "name", e.g., heart.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
 - "ssh heart.cs.rochester.edu" vs. "ssh 192.5.53.109"
 - <http://www.cnn.com> vs. <http://64.236.24.20>

10/28/2009

CSC 257/457 - Fall 2009

11

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, ...

10/28/2009

CSC 257/457 - Fall 2009

12

DNS: Decentralized and Hierarchical

- Decentralized
 - can't just have a single server/site to handle it
- Why not centralize DNS?
 - single point of failure, performance bottleneck
 - can only be close to few hosts ⇒ long latency to many others
- Hierarchical
 - there are different types of name servers
- What if massively replicating it?
 - **high overhead of maintaining consistency**: hard to make any changes on the data

10/28/2009
CSC 257/457 - Fall 2009
13

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 in its domain 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 in its domain 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 NSes
 - something is missing ...

10/28/2009
CSC 257/457 - Fall 2009
14

Root Name Servers

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

10/28/2009
CSC 257/457 - Fall 2009
15

DNS Query in Action

host **surf.kai.com** wants IP address of **xyz.cs.rochester.edu**

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.

10/28/2009
CSC 257/457 - Fall 2009
16

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

10/28/2009
CSC 257/457 - Fall 2009
17

Iterative Queries

recursive query:

- puts burden of name resolution on contacted name server

iterative query:

- contacted server replies with name of server for next contact
- "I don't know this name, but ask this server"

What is the real difference here?

10/28/2009
CSC 257/457 - Fall 2009
18

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 at end host)
- cache entries timeout (and removed) after some time
- timeout is necessary because 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

10/28/2009
CSC 257/457 - Fall 2009
19

DNS Properties

- Scalability
 - poor scalability at the few root name servers
 - made more scalable by caching
- Reliability
 - problems at (even a single) root name server(s) can cause big problem
 - again, caching helps improve reliability
- Consistency
 - DNS caching is replication without consistency guarantee
 - only feasible for stable name-IP address mapping, may not be suitable for more dynamic information


10/28/2009
CSC 257/457 - Fall 2009
20



TCP vs. UDP

- We saw applications:
 - Web/HTTP
 - Email/SMTP/POP3/IMAP
 - DNS
- They build on some transport services.
 - TCP or UDP?

10/28/2009 CSC 257/457 - Fall 2009 21



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).

10/28/2009 CSC 257/457 - Fall 2009 22