

Application-layer Protocols

Kai Shen

3/20/2013 CSC 257/457 - Spring 2013 1

Network Applications and Application-Layer Protocols

The diagram illustrates a network topology with various devices (laptops, servers, routers, mobile phones) connected. Red arrows indicate communication paths between these devices. Each device is associated with a vertical stack of protocol layers: application, transport, network, data link, and physical.

- 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

3/20/2013 CSC 257/457 - Spring 2013 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
 - request/response format

The diagram shows two client machines on the left and one server machine on the right. The top client is labeled 'PC running Explorer' and the bottom client is 'Linux running Firefox'. The server is labeled 'Server running Apache Web server'. Red arrows show 'HTTP request' going from each client to the server, and 'HTTP response' going from the server back to each client.

3/20/2013 CSC 257/457 - Spring 2013 3

Plain Text Headers

HTTP headers are plain text.
TCP/UDP/IP/Ethernet headers are binary.

Issues:

- Efficiency
- Portability
- Easy for human understanding, debugging

3/20/2013 CSC 257/457 - Spring 2013 4

Performance of HTTP Persistency

TCP per-connection overhead:

- Connection establishment
- Congestion control: slow start

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.

3/20/2013 CSC 257/457 - Spring 2013 5

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 content sharing (Gnutella, BitTorrent)
-

3/20/2013 CSC 257/457 - Spring 2013 6

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

3/20/2013 CSC 257/457 - Spring 2013 7

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 prot.

3/20/2013 CSC 257/457 - Spring 2013 8

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
  
```

3/20/2013

CSC 257/457 - Spring 2013

9

Try SMTP Interaction Yourself

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

This lets you

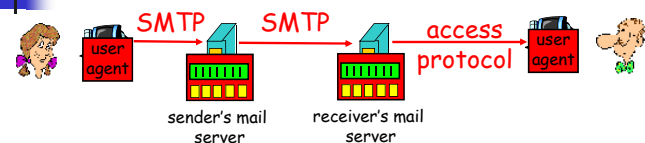
- send email without using a normal email client
- ...

3/20/2013

CSC 257/457 - Spring 2013

10

Mail Access Protocols



- SMTP: delivery to receiver's server and stored there
- 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.

3/20/2013

CSC 257/457 - Spring 2013

11

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 names?

- DNS query: find the IP address for a given name
- not part of Internet as a network
- ease of use for Internet applications
 - “ssh heart.cs.rochester.edu” vs. “ssh 192.5.53.109”
 - <http://www.cnn.com> vs. <http://64.236.24.20>

3/20/2013

CSC 257/457 - Spring 2013

12

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, ...

3/20/2013

CSC 257/457 - Spring 2013

13

DNS: Decentralized and Hierarchical

- Decentralized
 - a large number of servers/sites work together
- Why not centralize DNS?
 - performance bottleneck
 - single point of failure,
 - can only be close to few hosts ⇒ long latency to many others
- Hierarchical/partitioned
 - there are different types of name servers in charge of different mappings
- What if massively replicating it?
 - **high overhead of maintaining consistency**: hard to make any changes on the data

3/20/2013

CSC 257/457 - Spring 2013

14

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 ...

3/20/2013

CSC 257/457 - Spring 2013

15

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



3/20/2013

16

DNS Query in Action

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

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

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

3/20/2013 CSC 257/457 - Spring 2013 17

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

3/20/2013 CSC 257/457 - Spring 2013 18

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?

3/20/2013 CSC 257/457 - Spring 2013 19

DNS Caching

- Once a 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), but not at root name server
- 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


3/20/2013 CSC 257/457 - Spring 2013 20



DNS Properties

- Scalability
 - poor scalability at the few root name servers
 - made more scalable by iterative queries and 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


3/20/2013 CSC 257/457 - Spring 2013 21



TCP vs. UDP

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

3/20/2013 CSC 257/457 - Spring 2013 22



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).

3/20/2013 CSC 257/457 - Spring 2013 23