CSCE 463/612
Networks and Distributed Processing
Spring 2024

Application Layer

Dmitri Loguinov
Texas A&M University

February 2, 2024

Updates

```
http://x.com/path:900
http://x.com?script:900/
http://x.com?script/
http://x.com:8800?script:/
```

- URLs to try the parser on →
- Quiz 2: problems 5-33 end of Chapter 1
- Examine this fragment:

```
#define HUGE 10000000         // 10 MB
char buf [HUGE], *ptr = buf;
while((bytes = recv (sock, ptr, 100, 0)) != 0)
          ptr += bytes;

*ptr = NULL;
len = ptr – buf;
```

- Issues include
  - Inefficient recv
  - Buffer overflow when page exceeds 10 MB
  - Deadlock on errors
  - Deadlock if server doesn’t send any data
  - Probably stack overflow if buf declared in a function

Robots.txt

- Websites are crawled by many automated programs
  - This potentially consumes large volumes of traffic
- Besides bandwidth, concerns arise about protected or human-only portions of websites
  - Shopping carts, registration pages, posting into forums
- Webmasters need a mechanism to indicate prohibited request prefixes within their sites
  - These are given in /robots.txt

```
User-agent: *
Disallow: /search
Disallow: /sdch
Disallow: /groups
Disallow: /images
Disallow: /catalogs
Allow: /catalogs/about
Allow: /catalogs/p?
Disallow: /catalogues
```

- Directives are parsed in order, until first match
  - Algorithm has become ambiguous over the years: Google crawlers use the longest-prefix match

Robots.txt 2

- Despite being around since 1994, robots.txt is not a standard, but rather a suggestion on politeness
  - See http://robotstxt.org
- Extensions to robots.txt (even less official)
  - Crawl-delay specifies the # of seconds between visits
  - Sitemap points to an XML file that lists all available documents
  - Wildcards in directory paths (* and $ = ends with)

```
User-agent: *
Disallow: /*.asp$
Disallow: /sdch/*.php
Crawl-delay: 64
Sitemap: http://www.google.com/sitemaps_webmasters.xml
```

- How often should robots.txt be reloaded?
  - Original spec doesn’t say; Google uses 1 day by default

Chapter 2: Roadmap

2.1 Principles of network applications

2.2 Web and HTTP

2.3 FTP

2.4 Electronic Mail

- SMTP, POP3, IMAP

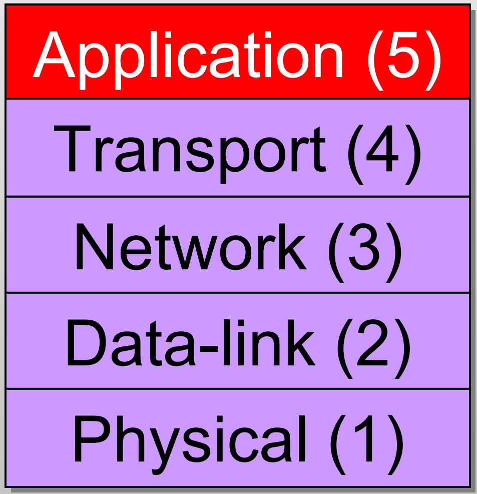
2.5 DNS

2.6 P2P file sharing

2.7 Socket programming with TCP

2.8 Socket programming with UDP

2.9 Building a Web server



Some Network Applications

- E-mail
- Remote login
- Web
- Instant messaging
- P2P file sharing
- Multi-user network games
- Streaming video
- Internet telephone
- Thermostat
- House alarm

- Real-time video conferencing
- Massively parallel computing
- Phones, tablets
- Internet fridge, TV

Mon, Jan 3, 2011 06:09 PM

Take out the garbage tonight

Do your homework

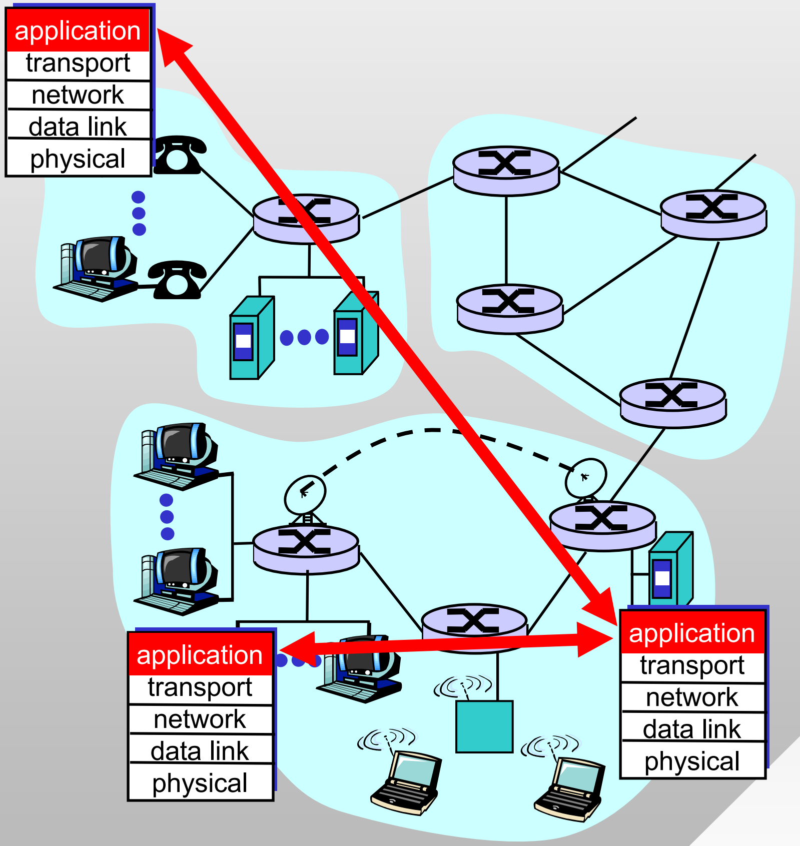
Creating a Network Application

Programs that

- Usually interact with user
- Communicate over a network
- E.g., Web server software communicates with browser software

No software written for devices in network core

- Network core devices do not function at app layer
- This design allows for rapid application development



Chapter 2: Roadmap

2.1 Principles of network applications

2.2 Web and HTTP

2.3 FTP

2.4 Electronic Mail

- SMTP, POP3, IMAP

2.5 DNS

2.6 P2P file sharing

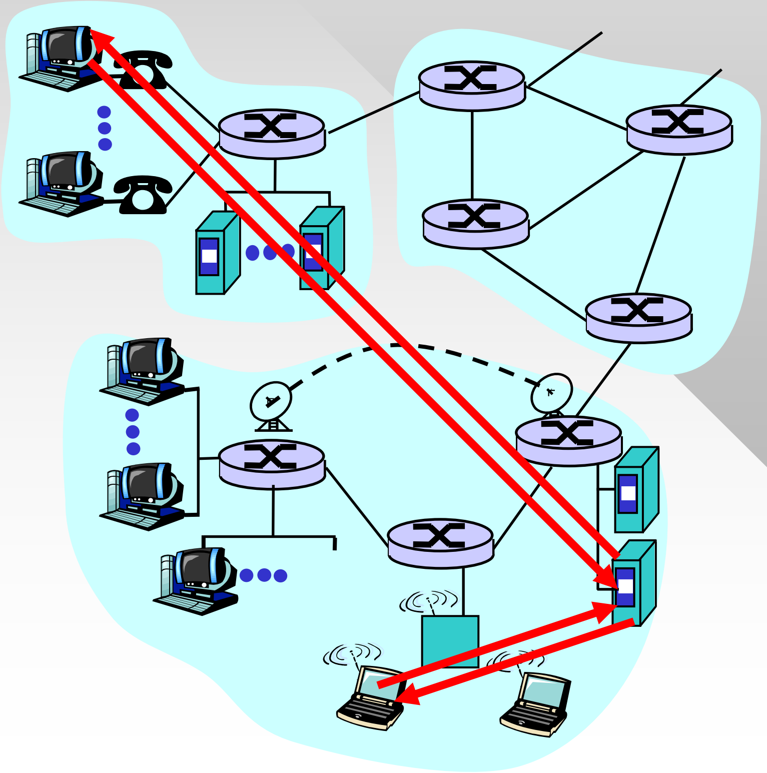
2.7 Socket programming with TCP

2.8 Socket programming with UDP

2.9 Building a Web server

Communication Principles

- Three architectures
  - Client-server
  - Peer-to-peer (P2P)
  - Hybrid



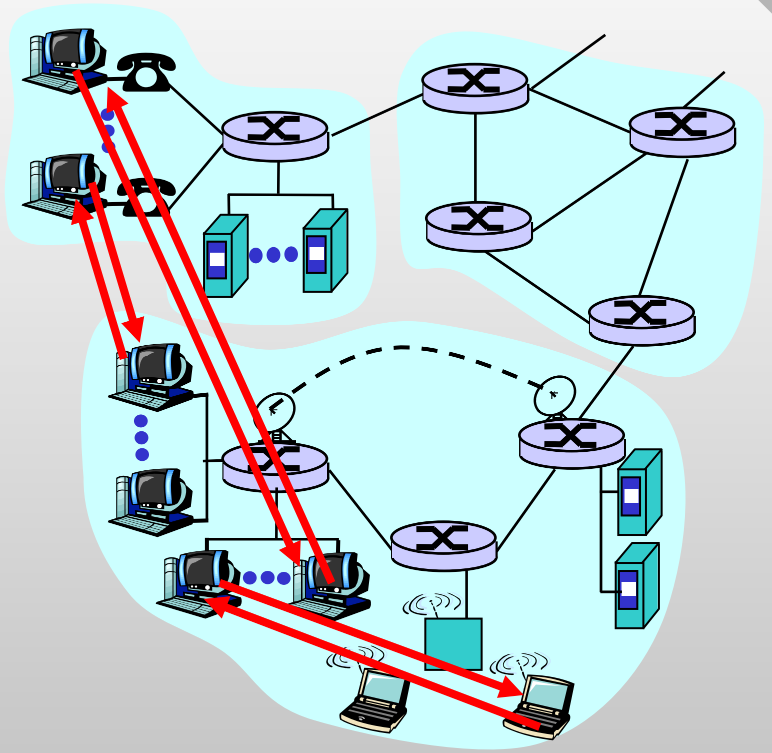
Server:

- An always-on host
- Permanent IP address or hostname
- Server farms for scaling

Clients:

- May be intermittently connected
- May have dynamic IP addresses and hostnames
- Do not communicate directly with each other, only talk to servers

P2P Architecture



- No always-on server
- Arbitrary end systems directly communicate
- Peers are intermittently connected and change IP addresses/hostnames
- Example: Gnutella
  - Distributed graph between users over TCP connections
- Highly scalable: assume 6M users with 1GB of shared data and 500 Kbps upstream bandwidth
  - 6 PB of storage, 3 Tbps bandwidth for free
- Downside – difficult to provide efficiency/reliability

Hybrid Architecture

Napster

- File transfer P2P, but search is centralized
  - Peers register content at central server
  - Peers query same central server to locate content

Instant messaging

- Login and chatrooms are centralized
  - User registers its IP address with central server
  - User contacts server to find IP addresses of friends or participate in chatrooms
  - But private chat is P2P (e.g., legacy Skype relayed data through other live peers)

Process Communication

- Process: program running within a host
- Within same host, two processes communicate using inter-process communication (semaphore, mutex, pipe, shared memory)
- Processes in different hosts communicate by exchanging messages

- Client: process that initiates communication
- Server: process that waits to be contacted

- Applications with P2P architecture act as both client & server