

Application Layer Introduction; HTTP; FTP

Tom Kelliher, CS 325

Feb. 4, 2011

1 Administrivia

Announcements

Assignment

Read 2.4-2.6.

From Last Time

Packet-switched network characteristics; protocol layers and models.

Outline

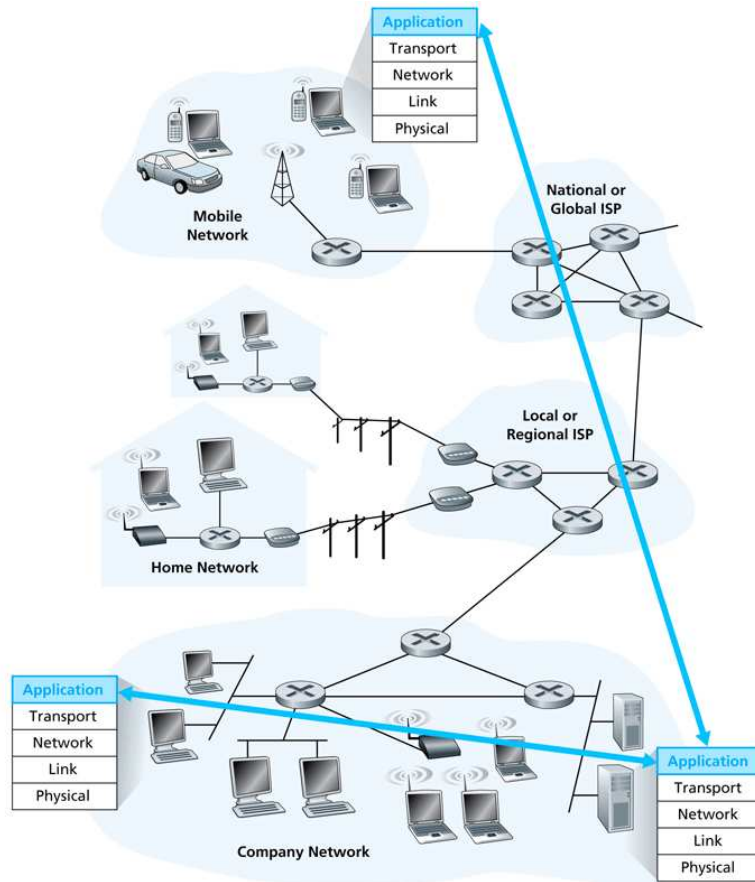
1. Introduction to the application layer.
2. HTTP protocol.
3. FTP protocol.

Coming Up

SMTP, DNS, and P2P.

2 Application Layer Introduction

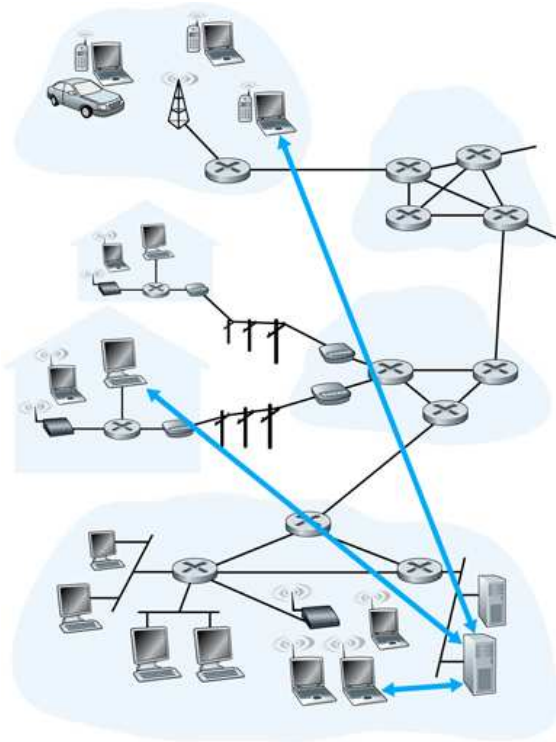
Internet applications have an abstract Internet view:



2.1 Internet Application Architectures

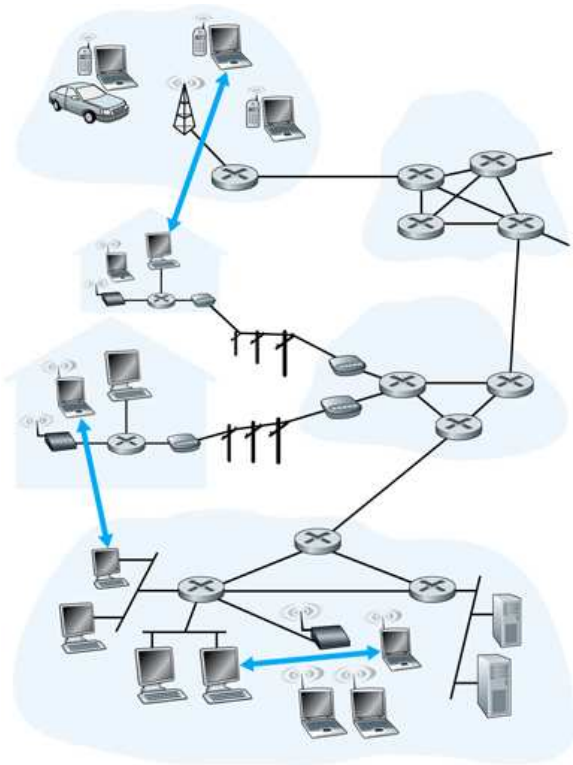
The Internet has an architecture; so do applications.

1. Client-Server architecture



- (a) Well-defined roles.
- (b) Infrastructure intensive — server farms. Expensive.
- (c) “Always on”
- (d) Hierarchical, centralized (ironic) — scalability problems, isolation vulnerabilities.
- (e) Examples: Google, Amazon, etc.

2. Peer-To-Peer Architecture

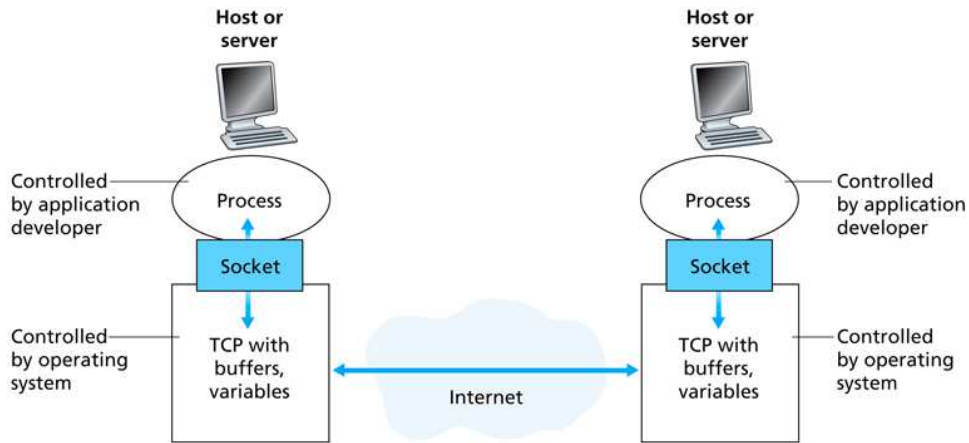


- (a) Roles not so well-defined.
- (b) Not infrastructure intensive. Cheap.
- (c) “Sometimes on.”
- (d) Flat, decentralized — scales well, less vulnerable.
- (e) May have some centralization features.
- (f) Examples: BitTorrent, LimeWire, Skype.

2.2 Process Communication

1. Request messages (from client), response messages (from server).

2. The Transport Layer provides the socket API:



2.3 Transport Service Properties

Two key Transport Layer protocols:

1. TCP is a connection-oriented service with reliable data transfer.

The connection must be established before messages can be sent.

Uses congestion control.

2. UDP is a connectionless service with no data transfer reliability — packets can be lost, arrive out of order, or be duplicated.

Just “throw” messages out there.

No congestion control.

Properties:

1. Reliable data transfer (TCP, not UDP).
2. Throughput (neither has any guarantees).
3. Latency (neither has any guarantees).

UDP has less overhead, an advantage.

4. Security (neither has security built in).

Layer SSL over TCP for security.

Streaming media uses buffering to mitigate some of these problems.

Realtime applications may use UDP to reduce latency.

Service access:

1. By IP address, port number.
2. A service on a host “listens” for service requests at a “well-known address” — its service port.

Example ports: 13, daytime (TCP and UDP); 20, ftp data; 21, ftp control; 25, SMTP; 80, HTTP; 443, HTTPS.

Properties of various network applications:

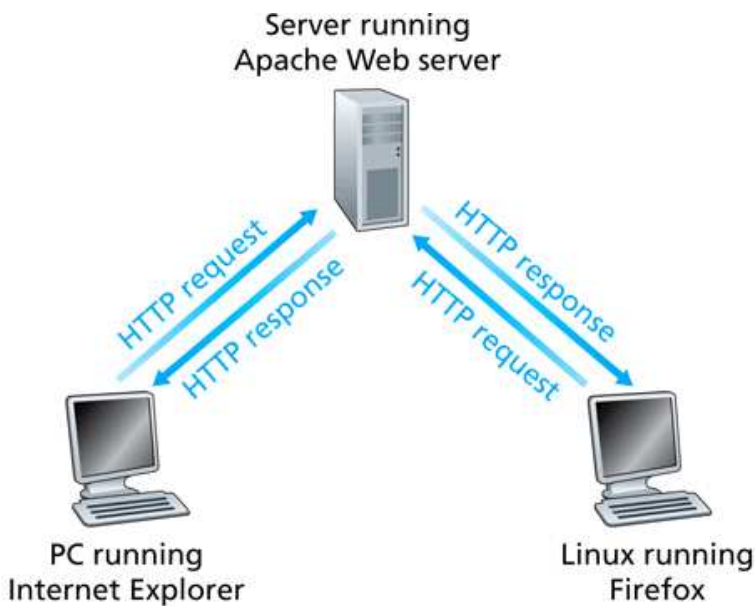
Application	Data Loss	Bandwidth	Time-Sensitive
File transfer	No loss	Elastic	No
E-mail	No loss	Elastic	No
Web documents	No loss	Elastic (few kbps)	No
Internet telephony/ Video conferencing	Loss-tolerant	Audio: few kbps–1Mbps Video: 10 kbps–5 Mbps	Yes: 100s of msec
Stored audio/video	Loss-tolerant	Same as above	Yes: few seconds
Interactive games	Loss-tolerant	Few kbps–10 kbps	Yes: 100s of msec
Instant messaging	No loss	Elastic	Yes and no

Protocol use of various network applications:

Application	Application-Layer Protocol	Underlying Transport Protocol
Electronic mail	SMTP [RFC 2821]	TCP
Remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
File transfer	FTP [RFC 959]	TCP
Streaming multimedia	HTTP (e.g., YouTube), RTP	TCP or UDP
Internet telephony	SIP, RTP, or proprietary (e.g., Skype)	Typically UDP

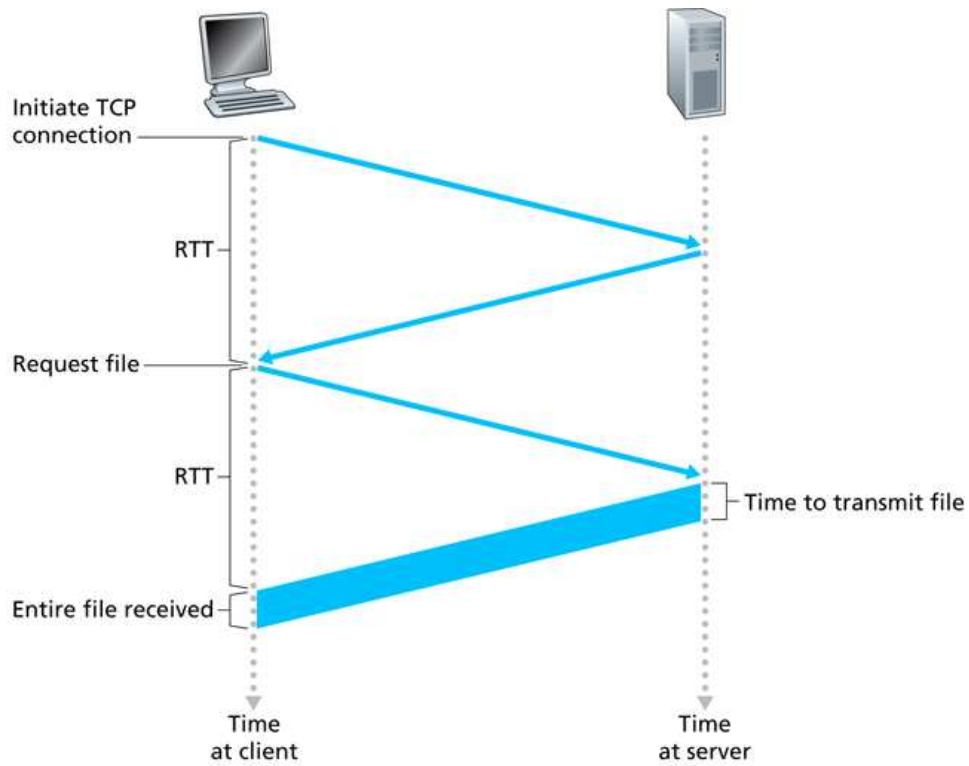
3 HTTP Protocol

1. HTTP uses client-server architecture:



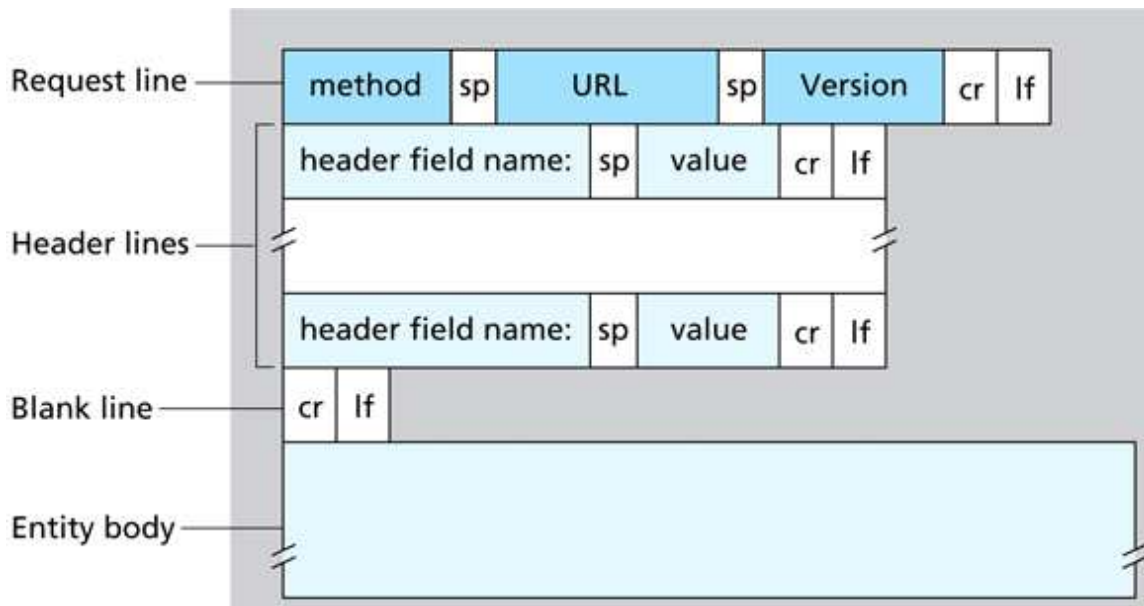
2. HTTP is stateless.
3. A single web page can contain several objects. How do we minimize load time?

Persistent and non-persistent TCP connections. Multiple, parallel TCP connections.

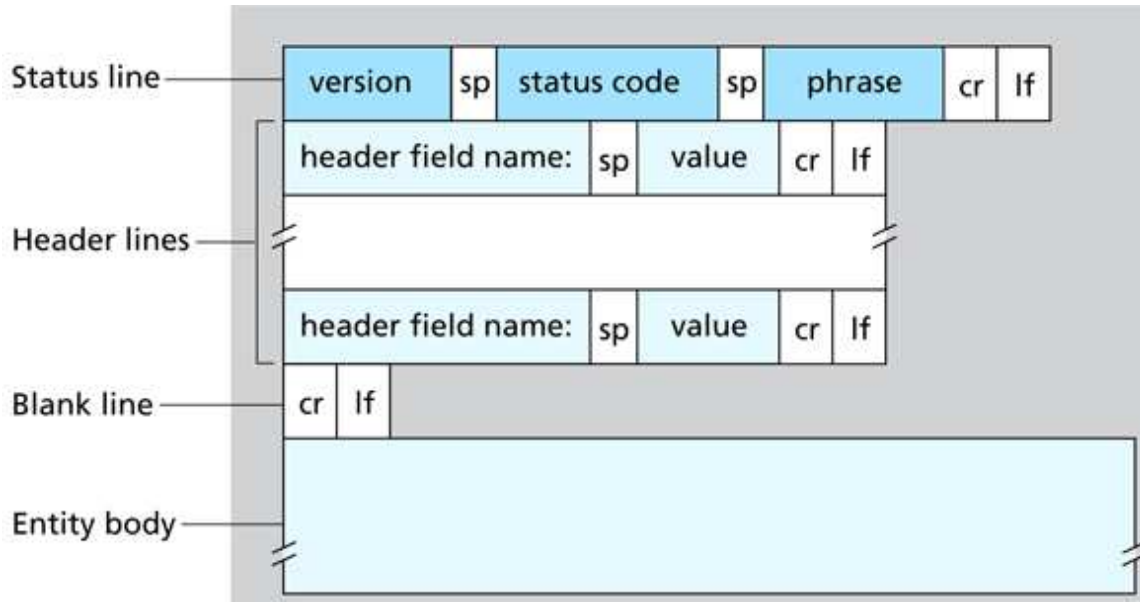


4. HTTP client methods: GET, HEAD, PUT, POST, DELETE.

HTTP client request format:



5. HTTP server response format:



6. Example exchange:

```

bluebird:~
% telnet phoenix http
Trying 10.32.3.39...
Connected to phoenix.
Escape character is '^]'.
GET / HTTP/1.1
Host: phoenix.goucher.edu

HTTP/1.1 200 OK
Date: Sun, 03 Feb 2008 20:40:29 GMT
Server: Apache/2.0.52 (Red Hat)
Accept-Ranges: bytes
Cache-Control: max-age=3600
Expires: Sun, 03 Feb 2008 21:40:29 GMT
Transfer-Encoding: chunked
Content-Type: text/html; charset=ISO-8859-1

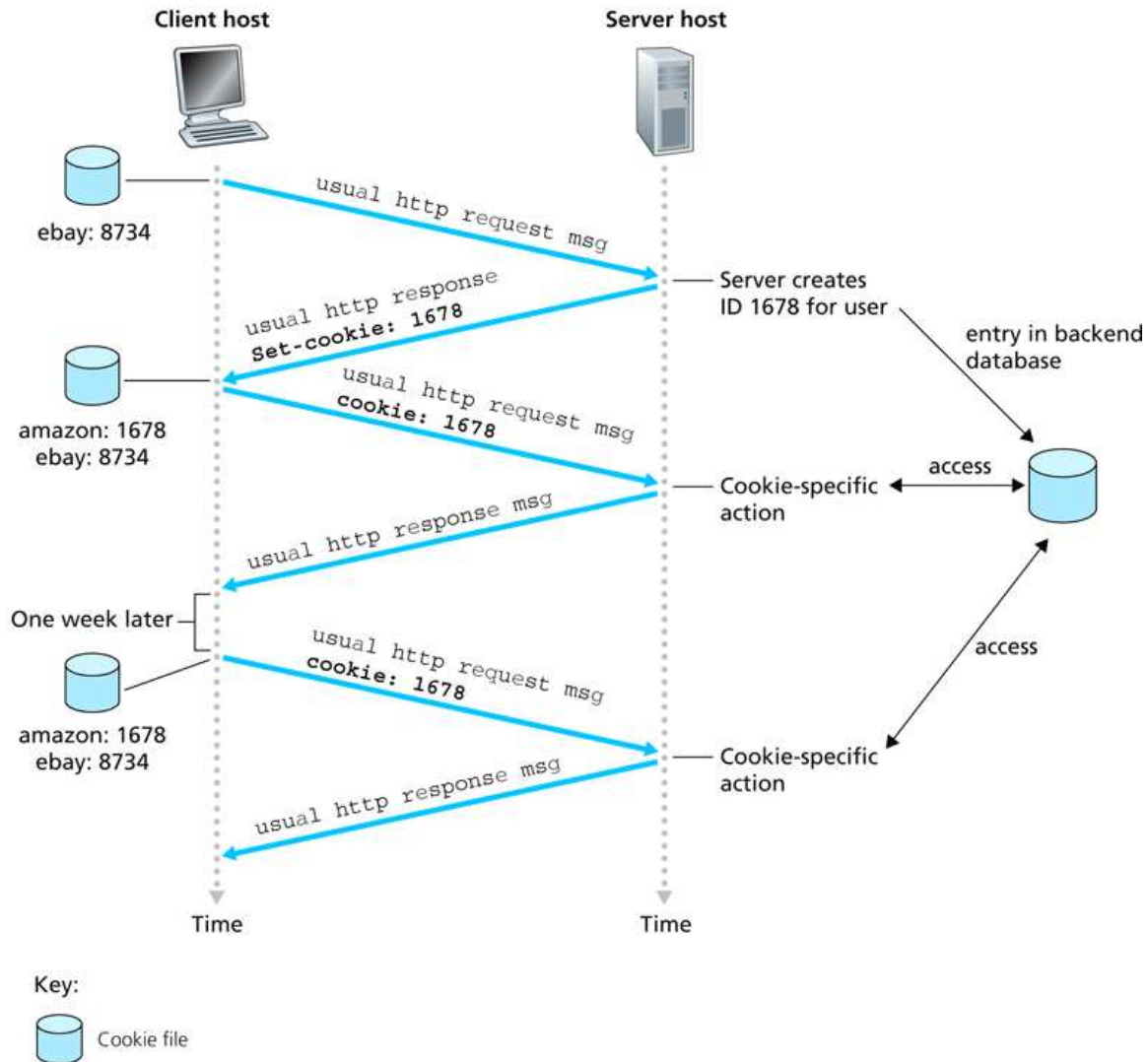
19f
<HTML>
<HEAD>
<TITLE>Welcome to Phoenix!</TITLE>
...
</BODY>
</HTML>

```

0

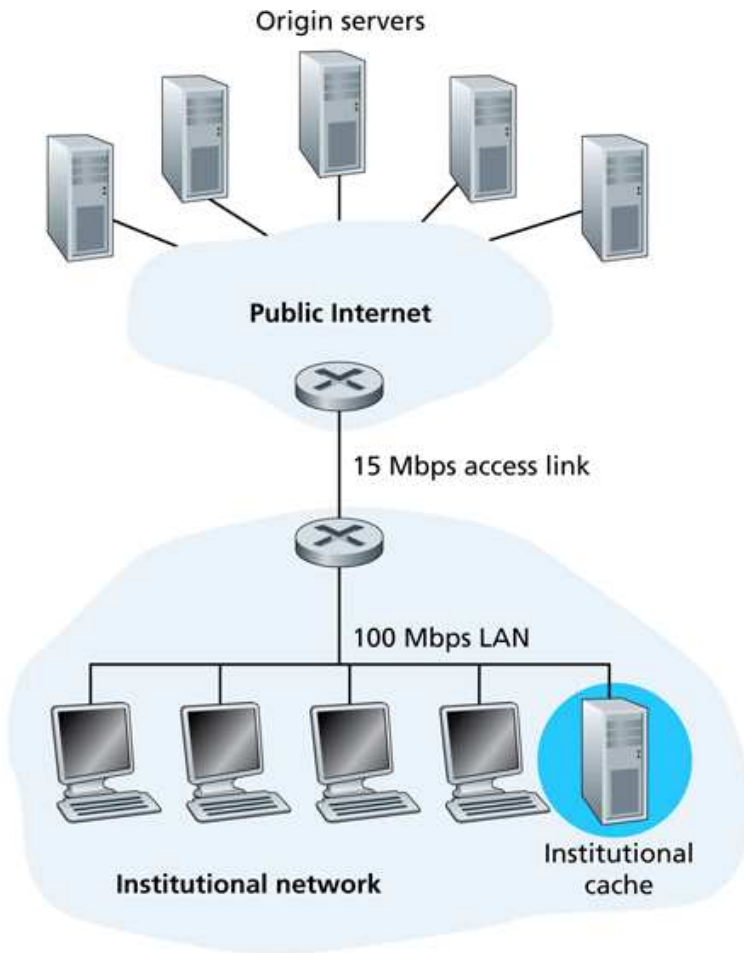
Connection closed by foreign host.

7. Cookies allow association with state information:



This provides a single- or multi-session layer on top of stateless HTTP.

8. Proxy servers can be used to lessen congestion at a bottleneck:

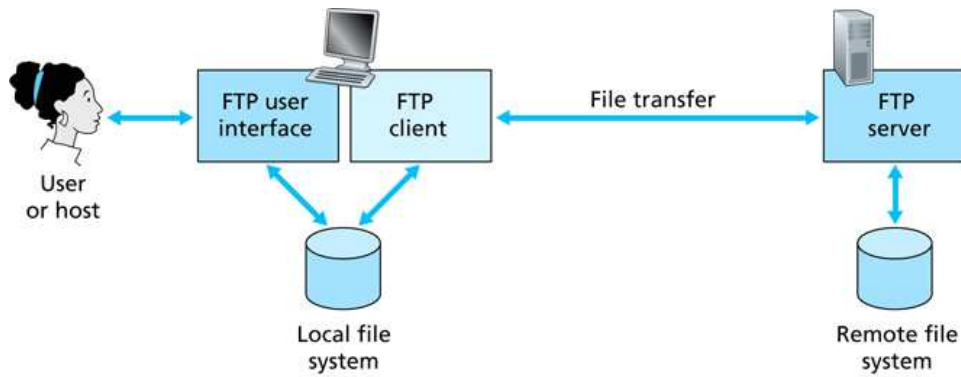


Client browsers are configured to send *all* requests to the proxy, which caches web objects. The proxy makes the real requests, on behalf of the clients.

4 FTP Protocol

HTTP and FTP are both basically file transfer protocols. What are the differences?

1. FTP elements:



2. FTP uses *out-of-band* control signaling:



3. FTP retains session state information: login, current directory, idle time.

More overhead because of statefulness

4. ASCII and binary data transfers.

5. Transfer modes:

- (a) Active: client opens a random data port, sends port number to server over control connection, which connects to client at given port (role reversal).

Problems result if there's a firewall at the client.

- (b) Passive: server opens a random data port, sends port number to client over control connection, which connects to server at given port number.