World-Wide-Web

- Tim Berners-Lee implements his childhood Enquire “2.0”
- It took off because it could do hypertext on the internet!
- 3 crucial components:
  - HTML, HTTP, URI
- A global documentation system at CERN [1990]
- As a way of justifying the purchase of a NeXT!
- First universal browser implemented by intern!

The Internet: A nuts and bolts view

- The Internet is a network of networks consisting of:
  - hosts;
  - communication links of varying bandwidths;
  - switching devices known as routers.
- Communication paths are shared using packet switching.

Hosts, clients, and servers

- Hosts can be can be either clients or servers.
- A client program running on one host requests and receives a service from a server program running on another system.
- (We also have P2P connections...)
Connection-oriented service ("TCP")

- **Reliable data transfer**: We can rely on the connection to deliver all of its data without error and in the proper order.

- **Flow control**: Neither side of a connection overwhelms the other.

- **Internet congestion control**: Attempts to prevent Internet gridlock.

Packet switching

- Messages are broken into packets each of which travels from the source to destination through a maze of routers and links.

- Most routers are **store-and-forward**, meaning the switch must receive the entire packet before it can transmit it outbound link.

There is also **Connectionless service ("UDP")**

- There are no handshakes with a connectionless service.

- The sending side simply sends the packets and hopes for the best.

- Useful in multimedia (e.g., phone, video) apps.

Message pipelining

- When the message is segmented into packets, the network is said to pipeline message transmission.
**The Internet backbone**

[Image of the Internet backbone]

**Horizontal layering of functionality**

<table>
<thead>
<tr>
<th>Layer</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ticket (purchase)</td>
<td>Ticket (complaint)</td>
</tr>
<tr>
<td>Baggage (check)</td>
<td>Baggage (claim)</td>
</tr>
<tr>
<td>Gates (load)</td>
<td>Gate</td>
</tr>
<tr>
<td>Runway takeoff</td>
<td>Runway landing</td>
</tr>
<tr>
<td>Airplane routing</td>
<td>Airplane routing</td>
</tr>
<tr>
<td>Departure airport</td>
<td>Arrival airport</td>
</tr>
<tr>
<td>Intermediate air-traffic control centers</td>
<td></td>
</tr>
</tbody>
</table>

**Taking an airplane trip**

[Diagram of an airplane trip]

**The Internet protocol stack**

- **Application layer** supports network applications (HTTP, SMTP, FTP).
- **Transport layer** provides transport of messages between client and server (TCP and UDP).
- **Network layer** is responsible for routing datagrams from one host to another (IP).
- **Link layer** moves frames from node to node (Ethernet, PPP).
- **Physical layer** moves individual bits of within a frame from node to node.
Client-server processes

- Programs don’t communicate, processes do.
- For each pair of communicating processes there is a client side and a server side.
- The host that initiates the session is labeled the client, while the process that waits to be contacted is the server.
- A process many play both roles at different times.

Process communication across the Net

- A process sends messages into, and receives messages from, the network through its socket or API.

The sending process must specify

- The name or address of the host (hostname or IP address), and
- The process on that host that will handle the received message (port number).

Hyper-Text Transfer Protocol

- HTTP is the Web’s client/server protocol.
- User agent (browser) implements the client side of HTTP.
- Web pages generally consist of a base HTML file which references other objects (JPEG, GIF, Java applet, audio clips).
HTTP/1.0 nonpersistent connection
www.someSchool.edu/someDepartment/home.index

1a. Client initiates a TCP connection to www.someSchool.edu on port 80.

2. Client sends HTTP request for /someDepartment/home.index to TCP socket set up in 1.

3. Server receives message through socket, finds, encapsulates, and sends object in HTTP response.

1a. Server at host www.someSchool.edu accepts connection and acknowledges.

5. Client receives response message. TCP connection terminates. Client extracts file, examines HTML, and requests other objects.

4. Server tells TCP to close TCP connection (but TCP waits to hear from client first).

6. First four steps are repeated for each referenced object.

The dance continues . . .

Timing of a nonpersistent connection

HTTP/1.1 persistent connections

- In an HTTP/1.1 persistent connection, the server leaves the TCP connection open after sending a response.
- Subsequent requests and responses (pipelined or not) between the same client can be sent on the same connection.
HTTP request message

GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close
Accept-language: fr

HTTP response message

HTTP/1.1 200 OK
Connection: close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 
Content-Length: 6821
Content-Type: text/html

Some HTTP response status codes

200 OK
request succeeded, requested object in this message
301 Moved Permanently
requested object moved, new location specified in this message (Location)
400 Bad Request
request message not understood by server
404 Not Found
requested document not found on this server
505 HTTP Version Not Supported

Testing HTTP using cURL

curl -v --http1.0 http://cs.wellesley.edu/~cs315 > response_body.txt
* Connected to cs.wellesley.edu (149.130.136.40) port 80 (0)
> GET /~cs315 HTTP/1.0
> User-Agent: curl/7.37.1
> Host: cs.wellesley.edu
> Accept: */*

< HTTP/1.1 301 Moved Permanently
< Date: Sun, 01 Feb 2015 14:08:47 GMT
< Server: Apache/2.2.15 (Red Hat) is not blacklisted
< Server: Apache/2.2.15 (Red Hat)
< Location: http://cs.wellesley.edu/~cs315/
< Content-Type: text/html; charset=iso-8859-1
< X-Varnish: 495673
< Via: 1.1 varnish-v4
< Content-Length: 322
< Connection: close
<
<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>322</td>
<td>100</td>
<td>322</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3126</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
* Closing connection 0