1.264 Lecture 16

Web introduction, HTTP

Next class: ASP.NET book chapters 2, 3. Exercises due after class
Exercise 1: HTTP request

Example browser request: (Browse to athena.dialup.mit.edu, log in)
telnet web.mit.edu 80 (telnet opens connection, http is port 80)
GET /admissions/ HTTP/1.1 (request)
Host: web.mit.edu (header)

(Blank line)

Example server response: (Hand in screen shot of response)
HTTP/1.1 200 OK (response)
Server: Apache/1.3.3 Ben-SSL/1.28 (Unix) (header)
Content-Type: text/html
Content-Length: 8300

(Blank line)

<HTML> (MIME (html) document)
<HEAD><TITLE>MIT…</TITLE></HEAD>
<BODY>MIT admissions…</BODY>
</HTML>

Connection closed (close connection)
Internet and Web Context

- We have covered databases, which are the nodes in our information network where data is stored.
- Internet and Web are the links that connect the nodes to transfer data from one node to another.
  - Web is one of many services that run over the Internet.
  - Internet is a system of interconnected computer networks.
  - Network has protocols to exchange data.
- Web is a system of hyperlinked documents.
  - Accessed using the Internet.
  - Referenced by Uniform Resource Locators (URLs).
- Web, for business, is a communications channel to transfer data from one database to another.
  - Documents can be dynamically generated, validated (using business rules) and exchanged.
  - Documents are often fragments of a database.
Internet and TCP/IP

• Internet is a set of loosely interconnected networks
  – Local area networks connected via wide area networks
  – No centralized control or management
  – Network segments interconnect via **routers:**
    • Dedicated computers that manage packets of data
  – TCP/IP is the universal **data** protocol on the network
    • TCP/IP: Transport control protocol/Internet protocol
    • **Content** is left to higher-level protocols, like the Web
    • **Transport** left to lower-level protocols, like Ethernet/WiFi

• TCP/IP connections are central to Internet and Web
  – Client is typically a data **consumer**
    • On Web, client is a browser or an application program
  – Server is typically a data **provider**
    • Web server listens for requests and transmits desired data
    • Web server gets data from application or database server
  – Web connection is active only long enough to exchange data
    • Avoid connection overhead, but lose session state
Transmission Control Protocol/Internet Protocol TCP/IP

Layer 7: Application
Layer 4: Transport
Layer 3: Network*
Layer 2: Data Link
Layer 1: Physical

Server computer

Layer 7: Application
Layer 4: Transport
Layer 3: Network*
Layer 2: Data Link
Layer 1: Physical

Browser or client computer

HTTP
TCP
IP
Ethernet
100BaseT

*-not same as layer 1/2 “network”
TCP/IP layering

TCP header: source port, destination port, seq nbr, checksum, time to live, out of band signal

IP header: protocol (TCP), IP source addr, IP dest addr

Ethernet header: Eth source addr, Eth dest addr, protocol (IP)

GET www.amazon.com

Layers

7: HTTP
4: TCP
3: IP
2: Ethernet
1: 100BaseT

Ethernet frame

Bits
Exercise 2: Web, TCP/IP, Ethernet

• Your laptop browser at IP address 18.9.3.2 sends
  – Web (HTTP) command on port 80: GET amazon.com
    • Other ports: ftp: 21, ssh: 22, telnet: 23, smtp:25, DNS: 53
  – To amazon’s IP address: 72.21.214.128
  – As a single IP packet at max time-to-live (255 hops)
  – TCP checksum computed to be 5324
  – Laptop Ethernet address is AA:BB:CC:DD:EE:FF
  – Router Ethernet address is 00:11:22:33:44:55
  – Ethernet checksum computed to be 6435

• Draw the Ethernet frame as it leaves your laptop with all field values filled in

• Draw the TCP packet received at the amazon server with all field values filled in
Solution

- **Ethernet frame sent from laptop:**

- **TCP packet at amazon server:**
  - 80 80 0 5324 254 0 GET amazon.com
  - (Time to live may be less than 254)
  - (Source port may be different than 80)
Web organization: Web pages

• Web page or document is unit of organization
  – Document is basic unit of business interactions
  – Pages have hypertext
    • Display text and images
    • Links to other pages, implemented via embedded URLs
  – Pages are described using Extensible Hypertext Markup Language (XHTML)
    • High level document description language
    • Specifies layout but not exact appearance of document
    • Browser handles display of page on client machine
  – XHTML has replaced HTML
    • Subset of XML, more structured, verifiable than HTML
    • XML is eXtensible Markup Language
Web organization: MIME types

• Every document on Web has a type
  – Multipurpose Internet Mail Extensions (MIME) types, in http standard
    • Some types, such as html, text and some graphics, are displayed directly by browser
    • Others need helper or plug-in external programs to display type
  – Web type system is extensible. New types are easy to accommodate.
    • Define new MIME type in server
    • Browser will prompt for helper application if MIME type unrecognized
    • Browser can suggest Web site to download helper app
    • Download and install helper app, and use the new MIME type
  – Web servers often generate documents dynamically
    • URLs point at programs as well as pages
    • Programs written in Java Enterprise Edition (JEE6), Microsoft .NET Framework, or LAMP (Linux-Apache-MySQL-php)
MIME type examples

- application/msword  Word
- application/pdf  Acrobat
- application/vnd.ms-excel  Excel
- application/zip  Zip file
- audio/basic   .au, .snd
- image/gif   GIF
- image/jpeg   JPEG
- text/plain   Plain text
- text/html   HTML
- text/xml   XML
- video/mpeg   Video

See http://www.rfc-editor.org for current list of HTTP, MIME, other Internet specifications
Hypertext transfer protocol: HTTP

• HTTP is core request-response protocol for Web
• Four phases:
  – **Open connection**: Based on URL
  – **Request**: Client opens connection to server and sends:
    • Request method, (and request data at bottom if POST or PUT request)
    • URL,
    • HTTP version number
    • Header information (informational, optional), terminated with blank line
  – **Response**: Server processes request and sends:
    • HTTP protocol version and status code
    • Header information, terminated by blank line
    • Text (data)
  – **Close connection**
These transactions are **stateless**. The connection is closed after each page and re-established: Server can’t connect successive requests from the client.

We use cookies and similar methods to simulate state.
## HTTP request phase (browser to server)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>Requests specified document (used to ‘post’ too)</td>
</tr>
<tr>
<td>HEAD</td>
<td>Requests only header of specified document</td>
</tr>
<tr>
<td>POST</td>
<td>Requests that server accept data from browser and generate dynamic content</td>
</tr>
<tr>
<td>OPTIONS</td>
<td>Get server and access options (OPTIONS *)</td>
</tr>
<tr>
<td>TRACE</td>
<td>Used in debugging</td>
</tr>
<tr>
<td>PUT</td>
<td>Replace server document with data from browser</td>
</tr>
<tr>
<td>DELETE</td>
<td>Delete specified document on server</td>
</tr>
<tr>
<td>CONNECT</td>
<td>Converts connection to tunnel, usually with SSL</td>
</tr>
<tr>
<td>PATCH</td>
<td>Applies partial modification to resource</td>
</tr>
</tbody>
</table>

- GET is the most frequent request from browsers
- POST used for processing forms (dynamic pages and input forms)
- HEAD used to check that content exists
- TRACE for debugging
- PUT, DELETE used for REST (Web service) middleware
- CONNECT used in secure connections
## Common HTTP request headers

<table>
<thead>
<tr>
<th>Header</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accept</td>
<td>MIME types accepted by client (multiple allowed)</td>
</tr>
<tr>
<td>Connection</td>
<td>Connection type for client (keep-alive, close)</td>
</tr>
<tr>
<td>Cookie</td>
<td>Return previous cookie to server (session data)</td>
</tr>
<tr>
<td>From</td>
<td>Email address of user (sent only by crawlers)</td>
</tr>
<tr>
<td>Host</td>
<td>Original host requested (forward, multiple names)</td>
</tr>
<tr>
<td>If-Modified-Since</td>
<td>Used to reduce fetching of docs client has</td>
</tr>
<tr>
<td>Referer</td>
<td>URL from which this link was obtained</td>
</tr>
<tr>
<td>User-Agent</td>
<td>Name and version of client software (browser)</td>
</tr>
<tr>
<td>Status code</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------</td>
<td>------------</td>
</tr>
<tr>
<td>100-102</td>
<td>Informational response; client should respond with some other action (continue, new protocol)</td>
</tr>
<tr>
<td>200-207</td>
<td>Request was successful</td>
</tr>
<tr>
<td>300-307</td>
<td>Document has moved; indicate new address</td>
</tr>
<tr>
<td>400-499</td>
<td>Client error, such as unauthorized request</td>
</tr>
<tr>
<td>500-510</td>
<td>Server error</td>
</tr>
</tbody>
</table>

**Examples:**

<table>
<thead>
<tr>
<th>Status code</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>OK</td>
</tr>
<tr>
<td>404</td>
<td>Not found</td>
</tr>
<tr>
<td>505</td>
<td>HTTP version not supported</td>
</tr>
</tbody>
</table>
## HTTP response header examples

<table>
<thead>
<tr>
<th>Header</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow</td>
<td>Requests allowed, such as GET</td>
</tr>
<tr>
<td>Content-Length</td>
<td>Length in bytes of data to be returned</td>
</tr>
<tr>
<td>Content-Type</td>
<td>MIME type of returned data</td>
</tr>
<tr>
<td>Expires</td>
<td>Date at which document expires</td>
</tr>
<tr>
<td>Last-Modified</td>
<td>Date at which document was last modified</td>
</tr>
<tr>
<td>Location</td>
<td>New document address (with 300 status)</td>
</tr>
<tr>
<td>Set-Cookie</td>
<td>Gives browser a ‘cookie’</td>
</tr>
</tbody>
</table>
Glossary

- **URL**: Uniform Resource Locator (Internet)
- **HTTP**: Hypertext Transfer Protocol (Web)
- **TCP**: Transport Control Protocol (Internet layer 3)
- **IP**: Internet Protocol (Internet layer 4)
- **100BaseT**: common Ethernet type (layer 2)
- **HTML**: Hypertext Markup Language (Web)
- **XML**: Extensible Markup Language (Web)
- **XHTML**: XML-based HTML (Web)
- **MIME**: Multipurpose Internet Mail Extensions
- **JEE6**: Java Enterprise Edition 6 (software)
- **.NET**: Microsoft software framework
- **LAMP**: Linux-Apache-MySQL-php (software)
1.264J / ESD.264J Database, Internet, and Systems Integration Technologies
Fall 2013

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.