Project 3: Basic Servers

- Telnet/Finger
- FTP
- Web
- SSH
- NNTP

Let’s look at the underlying protocols.
Client-Server Model

CLIENT

SERVER

Request

Response

Spring 2004
Client-Server Model

CLIENTS

SERVERS
Client-Server Model
Telnet
Telnet

- Remote terminal protocol.
- Allows users to login remotely.
Telnet

User’s machine

Telnet client

OS

TCP connection over Internet

Telnet server

OS

Spring 2004
Telnet basic operation

- When user invokes telnet, telnet client on user machine establishes TCP connection to specified server.
- TCP connection established; user’s keystrokes sent to remote machine.
- Telnet server sends back response, echoed on user’s terminal.
- Telnet server can accept multiple concurrent connections.
Handling heterogeneity

- Telnet protocol specifies standard data exchange: network virtual terminal (NVT).
- Telnet client and server make translation.
Finger

- Returns information on users currently logged on to a specified host.
- Finger server is often not running.
  - Implementation error in early version was used as the entry point for the "Internet worm" (1988).
  - Reveals private information.
Finger (cont’d)

- Also uses TCP.
- Client opens TCP connection to finger server.
- Client sends query.
- Server processes query, sends reply back, and closes connection.
- If finger server not running, client receives error message (e.g., “Connection refused”).
File Transfer

- Sharing remote files: “on-line” access versus “file transfer”.
- “On-line” access transparent access to shared files, e.g., distributed file system.
- Sharing through file transfer: user copies file then operates on it.
FTP

- File transfer accounted for most of the Internet traffic until the Web exploded!
- Also uses TCP.
- Allows interactive access; format specification (e.g., binary); authentication (clients required to authenticate themselves).

Spring 2004
FTP Operation

TCP connection
Port Assignment

- FTP server listens on well-known port (21); data transfer uses port 20.
- On client side, uses any unused port; client control process communicates that port number to server.
- Server process initiates data transfer connection.
Anonymous FTP

- Allows access to public files.
- No need to authenticate user.
- Access restricted to publicly available files (e.g., in /usr/ftp).
The Web and HTTP
The Web

- WWW, or the world-wide web is a resource discovery service.
  - Resource space is organized hierarchically, and resources are linked to one another according to some relation.
  - Hypertext organization: link “granularity”; allows links within documents.
  - Graphical user interface.
Some History

- Started in 1989 at CERN, European center for nuclear research, in Switzerland.
- Original motivation: need for scientists around the world to collaborate and share multi-media information.
- Tim Berners-Lee came up with initial proposal of a web of linked documents.
More History…

- First text-based prototype demo in 12.91.
- Release of first graphical interface, Mosaic, in 02.93 at NCSA by M. Andreessen.
- In 1994, Andreessen creates Netscape.
- In 1994, CERM and MIT set up the WWW Consortium to further develop the Web.
  - www.w3.org for more information.
The Client Side

- Users perceive the Web as a vast collection of information.
  - Page is the Web’s information transfer unit.
  - Each page may contain links to other pages.
  - Users follow links by clicking on them which takes them to the corresponding page.
  - This process can go on indefinitely, traversing several pages located in different places.
The Browser

Program running on client that retrieves and displays pages.

- Interacts with server of page.
- Interprets commands and displays page.

Examples: Mosaic, Netscape’s Navigator and Communicator, Microsoft Internet Explorer.

Other features: back, forward, bookmark, caching, handle multimedia objects.
The Server Side

- Web site has Web server running that answers requests for pages locally served.
  - Web server listens to port 80 for requests.
  - When request from client arrives, connection is set up.
  - Server replies.
  - Connection released.
Example

- User clicked on
  www.w3.org/hypertext/WWW/TheProject.html.
Example (cont’d)

- Browser asks DNS to resolve www.w3.org.
- DNS replies with 18.23.0.23.
- Browser sets up connection to 18.23.0.23 port 80.
- Browser sends GET /hypertext/WWW/TheProject.html.
- www.w3.org server sends TheProject.html file.
- Connection released.
- Browser displays TheProject.html, fetching and displaying all embedded objects (images, etc).
Observations

- Many browsers display status information at bottom of the screen.
- For each embedded object (in-line image like icon, picture, etc), browser establishes new connection.
  - Performance hit.
  - Revisions to protocol (HTTP) address this.
- Since HTTP is ASCII, easy for user to talk to Web servers directly (e.g., telnet to port 80).
More Observations...

- Server’s response specifies object type (using MIME) followed by object body.
- For example:
  - `Content-Type: Image/GIF`
  - `Content-Type: Text/html`
MIME

- Multipurpose Internet Mail Extensions (RFC 1521).

- Before MIME, objects (at the time e-mail messages) consisted of text only.
  - Only need to specify headers (subject, date, reply-to, etc).
  - Mainly due to Web, that no longer works: need to specify and structure object content.
MIME (cont’d)

- Adds structure to message body and defines encoding rules for non-textual messages.
- 5 new message (MIME) headers:
  - MIME version.
  - Content description.
  - Content id.
  - Content encoding.
  - Content type.
HTTP

- HyperText Transfer Protocol.
- Each interaction: client’s ASCII request followed by MIME-like response.
- Use TCP as underlying transport protocol (although not required by standard).
- Several co-existing versions of HTTP.
  - Newer versions specify protocol version in the request.
HTTP Operations

- Commands (method) to be executed on object (Web page).

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>Request to read Web page</td>
</tr>
<tr>
<td>HEAD</td>
<td>Request to read Web page’s header</td>
</tr>
<tr>
<td>PUT</td>
<td>Request to store Web page</td>
</tr>
<tr>
<td>POST</td>
<td>Append to specified object (e.g., Web page)</td>
</tr>
<tr>
<td>DELETE</td>
<td>Deletes Web page</td>
</tr>
<tr>
<td>LINK</td>
<td>Connects 2 objects</td>
</tr>
<tr>
<td>UNLINK</td>
<td>Disconnects 2 objects</td>
</tr>
</tbody>
</table>
HTTP Responses

- Every request gets response with status information.
  - Status code 200: OK.
  - Status code 400: bad request.
  - Status code 304: not modified.
Non-Persistent Connections

- Up to HTTP 1.0, separate connection for each data transfer.
  - Parallel connections.
Persistent Connections

- HTTP 1.1: persistent connections.
  - Same connection for multiple transfers.
  - Less overhead (connection management, slow start), less machine resources (buffers, connection id’s).
  - But, need to recognize beginning and end of an item (use length information).
Pipelining

- No pipelining: client issues new request only after getting response from previous request.
- Pipelining: client issues requests as soon as it finds a reference.
Proxy servers

- What are proxy servers?
  - Close to client.
  - Close to server.
Client-side proxies
Server-side proxies
Caching

- What’s caching?
- What are its benefits?
- What are its problems?
HTTP support for caching

- Servers can control caching.
  - Cachable/non-cacheable object.
  - Cacheable at proxy.
  - Cached object expiration time.
  - Operations performed on cached copy.
- Browsers can force request to go to server.
  - Specifies maximum age not to be > 0.
Internet caches

- Peers.
- Hierarchical caches.
Uniform Resource Locator

- Way to identify objects (pages).
  - What is page called?
  - Where is it located?
  - How to access page?

- URL has 3 parts:
  - Protocol (or scheme).
  - Machine’s name/address.
  - Local name (file name).
URL

- Ability to handle other protocols.
  - HTTP, FTP, news, gopher, mail, telnet.
HyperText Markup Language

- Allows users to produce Web pages including text, graphics, pointers, etc.
- Markup language: describe how objects are to be formatted.
  - Contains explicit commands for formatting.
  - Example: <B> and </B>.
  - Advantages: easy to parse.
Dynamic Pages

• A dynamic document is generated by the server at each new connection.
  – That’s why sometimes, when downloading from the same URL, we obtain different pages.

• Common Gateway Interface (CGI) standard defines server-application interaction.
  • CGI programs can be as simple as adding the time or date to the page.

• Browser may supply parameters to CGI program.
  – Browser extends URL with additional parameters separated by ?
Forms

- Forms permit a web page to have blank areas in which the user must enter information.
  - Makes it possible to enter data directly.
    - Name, address, credit card info…
  - Allows information to be sent to the server directly.
Forms

- HTML 1.0: one-way exchange.
  - Enable users to get information from servers.
  - Users could not send information back.
  - Example: on-line order forms.

- HTML 2.0 allows forms.
  - INPUT, SUBMIT.
  - `<FORM>` and `</FORM>`.
Common Gateway Interface.

- CGI scripts: programs that read input, process it, and generate output.
  - Example: after completing a form, user clicks on SUBMIT; this causes browser to invoke associated CGI script (given by URL). CGI script can generate output in the form of an HTML page that is returned to browser, which interprets and displays it.
  - Commercial use of the Web.
What is a Cookie?
What is a Cookie?

- A server invokes a CGI program each time a request arrives for the associated URL.
  - The server does not maintain any history of requests.
  - But a history is useful to allow CGI program to participate in dialog (e.g., to avoid having a user answer questions repeatedly).
  - Information saved between invocations is called state information.
  - State information is kept at the client’s side!
What is a Cookie (cont’d)

- State information is passed by browser in the form of a **cookie**.
  - The cookie is kept in the client’s computer.
  - When it contacts the Web server again, the browser inserts the cookie in the request.
A More Interactive Web

- HTML used to write static Web pages.
- With CGI, some “activation” possible.
- Java makes the Web more interactive.
  - Java program is executed directly on the client’s machine.
  - Allows to add animation, sound, etc. to Web pages efficiently
  - Remote execution model.
    - Instead of buying software, installing it, etc., just click on the appropriate applet from vendor’s Web page and get applet downloaded, executed, etc.
NNTP

- Network News Transfer Protocol.
- RFC 977 (1986)
- Distribution of news articles.
  - Inquire, retrieve, and post articles.
Alternative

- Mailing lists.
- Scalability issues.
- Maintenance issues.
(Usenet) News

- Client/server model.
- News server maintains news database.
- Client connects to server and reads, posts, etc. articles.
News Distribution

- News database distributed among several news servers.
- How to distribute news among servers?
- One alternative isflooding (UUCP)!
NNTP

- NNTP is “pull” based.
  - Server connects to its “neighbor”.
  - Asks whether there any new newsgroups.
  - Asks whether there are any new news articles in the relevant newsgroups.

- Uses TCP.
Some NNTP Commands

- **ARTICLE.**
  - Displays header and body of specified article.
- **HEAD and BODY.**
- **GROUP.**
  - Returns article numbers of first and last articles in the group.
- **IHAVE.**
  - Informs that client has articles specified.
More NNTP Commands

- LIST.
  - Returns list of valid newsgroups.

- NEWGROUPS.
  - Returns list of new groups created.

- NEWNEWS.
  - Returns list of new articles posted.

- POST.