Web Programming - 2
(Ref: Chapter 2)

- TCP/IP Basics
- Client Server Basics
- URL and MIME Types
- HTTP

Internet Architecture

Routers interconnect the network
TCP/IP software provides illusion of a single network

IP Software

IP software specifies how computers communicate, how a packet must be formed, how a router must forward each packet toward its destination.

IP datagram/packet -- Each packet sent across Internet follows the format specified by the IP.

For datagram routing ----- A unique address for a computer

IP Addressing

A unique 32-bit number for every computer ("Network Interface")

<table>
<thead>
<tr>
<th>Class</th>
<th>Prefix</th>
<th>Suffix</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1 1</td>
<td></td>
</tr>
</tbody>
</table>
### Dotted Decimal Notation

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>129.52.6.0</td>
<td>?</td>
</tr>
<tr>
<td>192.66.12.1</td>
<td>?</td>
</tr>
<tr>
<td>10.2.0.37</td>
<td>?</td>
</tr>
</tbody>
</table>

### IP Addressing

- **Class A**: 0 - 127
- **Class B**: 128 - 191
- **Class C**: 192 - 223

### IP Datagram/Network Packet

```
<table>
<thead>
<tr>
<th>IP Header</th>
<th>IP Data Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame Header</td>
<td>Frame Data</td>
</tr>
<tr>
<td>Source host</td>
<td>datagram</td>
</tr>
<tr>
<td>Router 1</td>
<td>datagram</td>
</tr>
<tr>
<td>Net 1</td>
<td>header 1</td>
</tr>
<tr>
<td>Net 2</td>
<td>header 2</td>
</tr>
<tr>
<td>Net 3</td>
<td>header 3</td>
</tr>
<tr>
<td>Destination host</td>
<td>datagram</td>
</tr>
</tbody>
</table>
```
TCP

TCP helps IP guarantee delivery
IP can drop datagrams if congestion occurs
Congestion?
If several computers trying to send data to a specific machine
Reordering of datagrams

Domain Names

Easy to remember -- Example: typhoon.cs.odu.edu
hostname.organization.organizationType
Translation of domain name to IP address
DNS -- distributed data base
DNS can be used to assign one or more aliases
Alias Machine
www.cs.odu.edu defiant.cs.odu.edu

Client Server Computing
(Distributed Computing)

Server --- service a request
Client -- makes a request

Ports

Port numbers to identify one of the several several programs a machine may be running
Standard port numbers to identify particular server programs

0 --- 65535
0-1024 on UNIX privileged

FTP 21
Telnet 23
Gopher 70
HTTP 80
NNTP 119
WAIS 210
Ports

client

192.23.43.114

cserver

23  Telnet

80  HTTP

70  Gopher

Daemons and Inetd

Daemons -- Stand alone servers (support multiple clients by forking --- Preforking )

Inetd -- to support several servers running on several ports

Inetd listens on every port and once receives a request on a port activate the appropriate port

Web Server--- mostly started as a daemon

URL

Uniform Resource Locator

communication protocol
name of the host machine
port number (if standard no need)
path

Example


http://bastet.cs.odu.edu:8000/custom/welcome.html

URL

Path in URL and UNIX

Web server interprets the URL path relative to the “document root” directory set in the server configuration

http://www.capricorn.org/cooking/curry.html

could point to

/local/web/cooking/curry.html

assuming /local/web is the document root directory
URL

Path name points to a file!
Path name points to a directory!
Path name points to a script (executable program)

http://www.capricorn.org/cgi-bin/phonebook

Path name to pass information to a script
http://www.capricorn.org/cgi-bin/phonebook?giles+goatboy

Path name for user supported directory
http://www.capricorn.org/~fred

MIME Types

Multipurpose Internet Mail Extension

Standard way of describing the type of documents

Type / Sub-type

Example: text/plain
text/html
image/gif
application/x-dvi (experimental x-)

Helper Application -- Use of external viewers to display files which can not be handled by the browser

File Extensions: .ps .gif .au

Virtual Hosts

Hosting several sites on a single servers

http://www.mall.com/capricorn/
http://www.mall.com/ferrets/
http://www.mall.com/zoo/

With the help of DNS alias

http://www.capricorn.org
http://www.ferrets.com
http://www.zoo.org

Servers need host name information with document request

Proxy Servers

To provide Web access to people located behind a corporate firewall.

Browsers connect to the proxy server and request a distant document --- Proxy server fetches it for them using special access privileges to cross the firewall

Proxy servers are also used for data cache.
HTTP Protocol

HTTP
telnet www.genome.wi.mit.edu http
Trying 18.157.0.107...
Connected to waldo.wi.mit.edu.
Escape character is ‘^]’.
GET /WWW/hello_daemon
Congratulations! If you see this you have
successfully had a two-way conversation with
a Web daemon!
Connection closed by foreign host.

HTTP
Protocol to support communication between browser and server

Request Phase

Methods:
GET  Return the content of the indicated document
HEAD Return the header information
POST  Treat the document as script and send some data to it
PUT   Replace the contents of the document with some data
DELETE Delete the indicated document

HTTP

Request Headers

From        Email address of the requesting user
Example --- From: zubair@cs.odu.edu
User-Agent  Name and version of client software
Example --- User-Agent: BrowserX 1.0
Accept      File types that client will accept
Example --- Accept: text/plain
Accept: text/html
Prioritize --
Accept: image/gif; q=0.5
Accept: image/jpeg; q=1.0
Accept: image/*; q=0.1
## Request Headers ....

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accept-Encoding</strong></td>
<td>Compression method that client will accept e.g. x-gzip, x-compress</td>
</tr>
<tr>
<td><strong>Accept-Language</strong></td>
<td>Language(s) that client will accept</td>
</tr>
<tr>
<td><strong>Referer</strong></td>
<td>URL of the last document the client displayed</td>
</tr>
<tr>
<td><strong>Authorization</strong></td>
<td>Method used along with the information</td>
</tr>
<tr>
<td><strong>If-Modified-Since</strong></td>
<td>Useful for caching</td>
</tr>
<tr>
<td><strong>Content-Length</strong></td>
<td>Length in bytes of data to follow</td>
</tr>
</tbody>
</table>

## Request Data

After request header and a blank line, the client can send data

## Response Phase

### Status Codes

- **200 - 299**: successful transaction
- **300 - 399**: document moved
- **400 - 499**: client made an error -- e.g. unauthorized request
- **>= 500**: server can not comply because of internal error

### Examples:

- **301**: Moved permanent -- new address in “Location”
- **302**: Moved temporary -- new address in “Location”
- **401**: Unauthorized -- authorization needed
- **402**: Forbidden

## Response Headers

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Server</strong></td>
<td>Name and version of server software</td>
</tr>
<tr>
<td><strong>Date, Last-Modified, Expires</strong></td>
<td>With 301 message -- new URL</td>
</tr>
<tr>
<td><strong>Pragma</strong></td>
<td>Hints to browser e.g. no-cache</td>
</tr>
<tr>
<td><strong>Content-Length, ContentType, Content-Encoding, Content-Language, Content-Transfer-Encoding</strong></td>
<td>With 405 (Methods not allowed) to inform client of all available methods</td>
</tr>
<tr>
<td><strong>WWW-Authenticate</strong></td>
<td>with 401 message</td>
</tr>
<tr>
<td></td>
<td>WWW-Authenticate: BASIC realm=&quot;Admin&quot;</td>
</tr>
<tr>
<td><strong>Allowed</strong></td>
<td>With 405 (Methods not allowed) to inform client of all available methods</td>
</tr>
<tr>
<td><strong>Set-Cookie</strong></td>
<td>Netscape specific extension</td>
</tr>
<tr>
<td><strong>Retry-After</strong></td>
<td>With 503 (service unavailable) date</td>
</tr>
<tr>
<td></td>
<td>Retry-After: 3600</td>
</tr>
</tbody>
</table>
Response Data

After the last header field the server sends an extra blank line. Then follows data (no data for the HEAD request method).

No limit on the size of data.

Normally connection closed after the transfer unless requested a persistent connection by sending a `Connection: Keep-Alive` header.