Week 6:
The World Wide Web

Johan Bollen
Old Dominion University
Department of Computer Science

jbollen@cs.odu.edu
http://www.cs.odu.edu/~jbollen
WWW: a short history

1. Predated by Gopher
   (a) Menus and links
   (b) Inter-server links
   (c) Largely defunct now
   (d) [http://www.faqs.org/rfcs/rfc1436.html](http://www.faqs.org/rfcs/rfc1436.html)
   (e) Try one:
      - `gopher://gopher.quux.org`
   (f) [http://www.scn.org/~bkarger/gopher-manifesto](http://www.scn.org/~bkarger/gopher-manifesto)

2. The WWW
   (a) 1989: developed at CERN
   (b) Original intent:
      i. Sharing physics data (text, images, files, etc)
      ii. Hypertext background: link content
      iii. Distributed means of storage
   iv. [http://www.w3.org/People/Berners-Lee/](http://www.w3.org/People/Berners-Lee/)

3. A timeline: [http://www.w3.org/History.html](http://www.w3.org/History.html)
   (a) 1989: formulation of protocols and markup language
   (b) 1992: mosaic browser
   (c) 1993: Netscape
   (d) 1995 Internet Explorer
Rapid Growth

Relation Between Percentage of Bytes & Packets Transferred Per Month By Service

Statistics provided by Merit MIC
Graphs by James Pitkow GYU
(pitkow@cc.gatech.edu)

Transition to new architecture starts to take effect
Prodigy et al. go online
How the WWW works

Different Perspectives

1. Client-server architecture
   (a) Server: contains web pages and related files
   (b) Client: request pages, visualizes and generates user interface

2. Four functional components
   (a) Client-Server File transfer: HyperText Transfer Protocol
   (b) Presentation: Web browser/Client
   (c) Page Identification: Universal Resource Locator
   (d) Page specification: HyperText Markup Language
WWW servers (HTTP)

1. Application that runs on any particular internet host
2. Listen on port 80
3. Respond to HTTP requests (more later)
4. Can serve a variety of files and applications
5. Common types:
   (a) Apache (opensource):
       http://httpd.apache.org/
   (b) Microsoft
   (c) NSCA
   (d) SunOne

October 7, 2003
HTTP servers marketshare
Universal Resource Locators

Identification of web resources

1. Each page must have unique ID
   (Universal Resource Identifier)

2. Backwards compatible with older protocols
   (a) FTP
   (b) Gopher

3. respect existing filesystem standards
   (a) Assumption: host + path + file name
   (b) Host: internet host name
   (c) Path: server root + subdirs
   (d) Filename: any that identifies WWW page
URL format

http://www.cs.odu.edu/~jbollen/cs312.html

protocol  hostname  path filename
URLs

1. Four components
   (a) Scheme
      i. HTTP
      ii. FTP
      iii. gopher
      iv. mailto
   (b) hostname
      i. host = server
      ii. IP number will do
   (c) path
      i. relative to server root
      ii. absence of path: default

2. Absolute addressing for web pages
   (a) No inherent verification
   (b) No mechanism for link maintenance
   (c) Duplication? When are two pages the same?
URLs: character encoding

1. Subset of ASCII set
   (a) Alphanumerical
   (b) $-_.+!*’()$
   (c) what about other characters? e.g. @ (mailto)

2. Character encoding: which characters
   (a) Control Characters (non-printable)
   (b) Non-ASCII
   (c) Reserved: normally part of URL

3. HEX coding
   (a) % symbol, followed by the two-digit hexadecimal representation
   (b) for example: tilde: %E7
   (c) http://www.blooberry.com/indexdot/html/topics/urllencoding.htm
HTTP: HyperText Transfer Protocol

1. Client-server communication
   (a) Stateless
   (b) Aimed at simple pull-type transactions
   (c) In-band transfer
   (d) Verbs and returned information: ASCII

2. Port

3. Verbs
   (a) GET
   (b) HEAD
   (c) POST
   (d) PUT

4. Server responds:
   (a) 200: OK
   (b) 400: Bad request
   (c) 404: Not found
   (d) Or: HTML page

5. http://www.w3.org/Protocols/HTTP/
HTTP: telnet to port 80

http://www.cs.odu.edu/~jbollen/telnetp80.png
http://www.cs.odu.edu/~jbollen/CS312/telnetp80_head.png
http://www.cs.odu.edu/~jbollen/CS312/telnetp80_error.png
HTTP and the Web: some peculiarities

1. What’s index.html?
   (a) Absent filename
   (b) Server by default returns index.html
   (c) this allows: 
       http://www.cs.odu.edu/

2. Why can I sometimes not complete URL or malform them?
   (a) Client fills in blanks
   (b) be careful though

3. What are those long URLs with lots of ampersands
   (a) Note question mark
   (b) Information is passed back to server
   (c) packed in URL
HTTP: some notes

1. Free for all
   (a) No password protection or authentication
   (b) Enables access for all that speak HTTP
   (c) Simple, efficient scheme to distribute data
   (d) Possibility of two-way traffic (not duplex!)

2. Possibility of layering
   (a) Protocols on top of protocols
   (b) Use HTTP to transfer commands associated with meta-protocol
   (c) Data transferred in HTML or any other format

i. URL encoding
ii. POST
iii. PUT