World Wide Web - History, Architecture, Protocols
Architecture of Web Information Systems

CS/INFO 431
Carl Lagoze - Spring 2006
Creating Order from Chaos

• Information universe is inherently disordered

• Cognition is order-making, pattern finding
  - Hawkins - “On intelligence”
  - Classification
  - Data mining

• Information management involves, then, putting layers of order on this chaos
  - policies, practices, standards, laws, architectures
Standards in traditional information management

• Evolved in slow transition from elite culture to democratic culture

• Professional Culture controls adaptation
  - Shared culture through professional affiliation, ALA, IFLA
  - Shared culture through training, MLS

• Codes
  - Library Bill of Rights
  - Privacy agreements

• Intellectual Standards
  - Dewey Decimal System
  - Taxonomies - LCSH, MESH
  - Cataloging Rules - AACR2, Name Authorities

• Architectures
  - Machine Readable Cataloging
Standards in networked information management

• Roots in elite culture, revolutionary transition to democratic culture
• Complicated by profit/power potential
  - Political structures reflect this complication
• Based on code rather than human behavior
  - difficult transition from heuristic to algorithmic world – e.g., rights management
  - Larry Lessig “Code and Other Laws of Cyberspace”
• Opportunities to replace human effort with algorithmic and computational power
• “Good enough” principle
## Architecture and Standards Layers

<table>
<thead>
<tr>
<th>Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Semantics – DTD, Schema, RDF, OWL</td>
</tr>
<tr>
<td>Web Protocols and Standards – XML, HTTP</td>
</tr>
<tr>
<td>Internet - TCP/IP, SMTP, email, etc.</td>
</tr>
<tr>
<td>Network Hardware</td>
</tr>
</tbody>
</table>

Upper layers operate within constraints and opportunities of lower layers
In the beginning....
In the beginning...
ARPANET

- DoD funded through leadership of Licklider
- Inspired by move from batch to timesharing
- Allowed remote login
Packet Switching

- Invented in early 1960’s by Baran, Davies, Kleinrock
- Digital, redundant, efficient, upgradeable (software)
- 1969 ARPANET first network implementation
Packet Switching

• Network messages broken up into packets
• Each pocket has a destination address
• Pass and forward model – router gets packet, examine, decides where to send next
• Message reassembled on other end
Layered Protocol Model

- **Process**: User applications
- **Transport**: Host-to-host layer: may establish virtual circuit for messages
- **Network**: Determines route to transmit packets
- **Data Link**: Transforms raw messages into one that appears free of transmission errors, transmits data cells and processes acknowledgement cells.
TCP/IP Protocol Suite

- **IP** - packet delivery
- **TCP** - virtual circuits, packet reassembly
- **ARP/RARP** - address resolution
Internet Issues – how to address them

- Demands of multimedia applications
- Virtual circuit reservations - bandwidth and quality of service guarantees
- Real time streaming protocols
- State saving

- Political Comment
  - Increase in functionality has implications
    - Democratization of the Net
    - Privacy
    - Vulnerability
THE FUTURE OF IDEAS

THE FATE OF THE COMMONS
IN A CONNECTED WORLD
Infrastructure and Standardization

• Complex legal, economic, social, and technical process
• Wasn’t invented in the information age
  – Railroad track gauge and tariffs
  – Telephone and telegraph
  – Banking
  – Power and Light
• Not for the faint-hearted
Internet Governance

- Internet Society (ISOC) - Evolution, social & political issues
  - http://www.isoc.org/
- Internet Architecture Board (IAB) - Oversees standards process
  - http://www.iab.org/
- Internet Engineering Task Force (IETF) - standards development
  - http://www.ietf.org/
- Internet Corporation for Assigned Names and Numbers (ICANN)
  - DNS administration
  - IP # assignment
  - Protocol #’s
  - port #’s
  - http://www.icann.org/
- World Wide Web Consortium (W3C) - web standards and evolution
  - http://w3c.org
Internet Documents

- RFC’s – “Requests for Comments” to IETF community for information, standardization
- STD’s – Official IETF Internet standards
  - [http://www.rfc-editor.org/rfcxx00.html](http://www.rfc-editor.org/rfcxx00.html)
- Internet Drafts – IETF working documents
- W3C Reports (recommendations, drafts, notes)
  - [http://www.w3.org/TR/](http://www.w3.org/TR/)
Well-Known Protocols

- Telnet - external terminal interface, RFC 854 (1983)
- FTP - file transfer, RFC 959 (1985)
- SMTP - mail transport, RFC 821 (1982)
- HTTP - distributed, collaborative hypermedia systems, RFC 1945 (1.0 1996), RFC 2616 (1.1 1999)
Short History and Premises of the Web

• Information sharing in a fluid context
  - CERN 1989
  - Reality
• Relationships are not hierarchical
• Non-centralized management
• Structure can be modeled as a graph
  - Typed nodes (text, graphics, people, software modules)
  - Type relationships (depends on, refers to, made)
• Hypertext (after Ted Nelson)
  - Human-readable information linked together in an unconstrained way.
  - Extend to Hypermedia and network
• Clean division of document display and format (browsers and HTML) from access (HTTP)
Basic Web Technologies

- Document layout
  - HTML → XML
- Document formatting
  - CSS
- Document naming
  - URL’s
- Document typing
  - MIME
- Document access
  - HTTP
HTTP

• HTTP is...
  - Designed for document transfer
  - Generic
    • not tied to web browsers exclusively
    • can serve any data type
  - Stateless
    • no persistent client/server connection
  - Defined at ftp://ftp.isi.edu/in-notes/rfc2616.txt
HTTP Example

[-bash-2.05b4] telnet google.com 80
Trying 72.14.207.99...
Escape character is '^]'.
GET index.html HTTP/1.1
Host: lagoze.com

HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html
Set-Cookie: PREF=ID=9c9f0e0565b57456:TM=118635078:LM=118635078:S=H-BswXLg53YkL
114; expires=Sun, 17-Jan-2038 19:14:07 GMT; path=/; domain=.google.com
Server: GWS/2.1
Transfer-Encoding: chunked
Date: Mon, 30 Jan 2006 15:31:18 GMT

bd8
<html><head><meta http-equiv="content-type" content="text/html; charset=ISO-8859-1"><title>Google<title><!--body,td,a,p,.h{font-family:arial,sans-serif;}
.h{font-size: 20px;}
.q{color:#0000cc;}
//-->
</style>
<script>
<!--
function sf(){document.f.q.focus();}
//-->
</script>
</head><body bgcolor=#ffffff text=#000000 link=#0000cc vlink=#551a8b alink=#ff00
00 onlload=sf() topmargin=3 marginheight=3><center><table border=0 cellspacing=0
cellpadding=0 width=100%><tr><td align="right" nowrap><font size=-1><a href="/url?q=http://www.index.html/1g%3Fhl%3Den">Personalized Home</a></font></td><tr height=4><td><img alt="" width=1 height=1></td><tr><td><table cellpadding=0 cellspacing=0 border=0><tr><td align="right" valign="bottom"></td>
HTTP Session

• An HTTP session consists of a client request followed by a server response
• Requests and responses are sent in plain text
HTTP Request Methods

• Methods include
  - GET: retrieve information identified by the URL
  - HEAD: same as get but don't get message body (content)
  - POST: accept the request content and send it to the URL
  - PUT: store the request content at the given URL
HTTP Request

• Start line
  - Consists of method, URL, version
    
    `GET index.html HTTP/1.1`
  - Valid methods include:
    - `GET`, `POST`, `HEAD`, `PUT`, `DELETE`

• Headers
  - `HTTP/1.1` requires a `Host: header`
    
    `Host: www.google.com`

• Body content
HTTP Response

• **Start line**
  - consists of HTTP version, status code, and description
  HTTP/1.1 200 OK
  HTTP/1.1 404 Not Found

• **Headers**
  Content-type: text/html

• **Content**
HTTP Response Codes

- Response coded by first digit
  - 1xx: informational, request received
  - 2xx: success, request accepted
  - 3xx: redirection
  - 4xx: client error
  - 5xx: server error

HTTP Content Body

• Header fields can affect content interpretation
  - required header field: Content-type
  - others: Content-Encoding, Content-Length, Expires, Last-Modified