ECE 435 – Network Engineering
Lecture 3

Vince Weaver

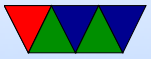
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

5 September 2017

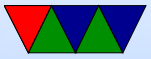
Announcements

- Homework #1 was posted.
- `strace` can be useful when tracking down issues.

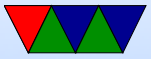


The World Wide Web (history)

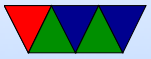
- Before: getting files via cd-rom or ftp (or e-mail/ftp gateways!), search with archie (archive w/o the V, not comic related)
- gopher: university of Minnesota, 1991. search with jughead/veronica
  Why fail? UMN tried to charge license fee, much more restricted file format than html.
- World-Wide-Web: Tim-Berners Lee, CERN, Initial Proposal 1989, first text-based prototype 1991



- Marc Anderson UIUC worked on graphical browser, Mosaic, 1993
- Anderson went on to form Netscape Communications 1994. Webserver software, made Navigator ("mozilla") relatively cheap/free to drive uptake of web servers.
- Microsoft Internet Explorer. Licensed version of Mosaic. 1995 (as add-on to Win95). MS paid percentage royalties to Spyglass Mosaic, so what happened when they gave it away for free?
- Browser wars.
- Netscape bought by AOL in 1998



- By 2000, IE had over 80% due to bundling with windows, famous lawsuit
- Gap between IE6 and IE7 of 5 years (2001 to 2006)
- Netscape released firefox as open source in 2004
- Safari/Webkit browsers based off of KDE browser
- Google Chrome took over the lead around 2012 or so
- Standards fight. ACID test.

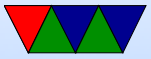


Top Browsers

1996	Mosaic 1.2%	Netscape 77.3%	IE 19%				
2003	IE 94%	Firefox 2%	Safari —	Opera 1%	Navigator 1%		
2010	IE 42%	Firefox 29%	Chrome 11%	Safari 6%			
2017	Chrome 46.5%	Safari 21.5%	IE 10.1%	Firefox 6.3%	Edge 1.9%	Opera 1.3%	Android 1.2%

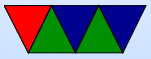
Stats from Wikipedia. EWS for 1996. TheCounter.com for 2003, wikimedia 2010,2017

Other browsers: midori, lynx, links, w3m

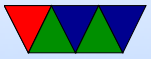


HTML

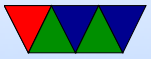
- HTML – hyper text markup language
- Based on SGML (Standard Generalized Markup Language)
- Hypertext (documents that can link to each other) actually proposed by Vannevar Bush in 1945
- Simplest form, just a text file with some extra commands specified in angle brackets, and usually a closing tag with a / in it. Case insensitive (though supposed to use lowercase these days).



- Standards
  - Internet Engineering Task Force (IETF) HTML 2.0 in 1994
    RFC 1866, 1867, 1942, 1980, 2070
  - Since 1996 by the World Wide Web Consortium (W3C)
  - 2000 HTML (ISO/IEC 15445:2000)
  - HTML 4.01 in 1999
  - HTML5 by Web Hypertext Application Technology Working Group in 2014
  - Javascript (ECMAscript) ECMA-262 and ISO/IEC 16262

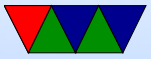


- HTML4 vs HTML5 vs XHTML
- XML extensible markup language, can do things like add new tags on fly



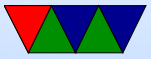
Sample ancient HTML

```
<!DOCTYPE html>
<html>
<head><title>ECE435 Test</title></head>
<body>
<center><h1>ECE435 Test</h1></center>

<hr>

This is a test.
<img src="image.png" alt="alternate text">
```

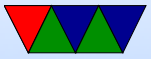


```
<br>Line Break
<!-- Comment -->
<p> Paragraph
<b>Bold</b> <i>Italic</i>

<a href="other.html">A link to another page</a>

</body>
</html>
```

- Tables also easy to do.

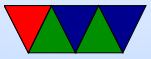


- Early on vendors went crazy with custom tags: Marquee tag, Blink Tag. Frames.
- “view source”
- Originally idea was no formatting, web browser should automatically display simple text in a way to best be displayed on your local machine
  Publishers/graphics designers got a hold of it and that’s where all the pixel perfect positioning stuff came in
- CSS (cascading style sheets), Javascript
- Submitting back to the website, HTML forms

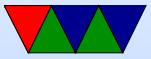


Dynamic Content

- Server Side
  - cgi-bin: Write a program that takes input as environment vars, output as standard out sent to the requesting browser.
    Can write in any program. Typically was things like perl, I often did this in C or even Fortran
  - Dynamic content – SSI (server side includes)
  - Server extensions (such as PHP, modperl, ASP, .NET) more commonly used (with security issues)

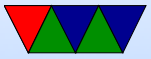


- Client Side
  - Javascript horror. Client side, code runs on your computer rather than on the server.



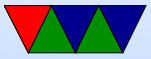
www on the client side

- A URL (uniform resource locator) specifies the document you want
  `http://web.eece.maine.edu/~vweaver/`
- Browser parses URL
- It looks up the address of web.eece.maine.edu via DNS
- DNS reports 192.168.8.99
- Browser makes TCP connection to port 80 (default) of 192.168.8.99
  how do you specify other port? web.eece.maine.edu:8080



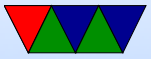
why would you want to?
security

- The client requests the file vweaver/index.html (index.html is the default). Often there's a server root, and special handling for user dirs (with the tilde) that are often in user home dirs, as in this case
- The server returns this file
- The TCP connection is closed
- The browser displays the text
- The browser might have to fetch any images linked to by the document



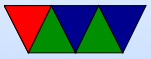
Non-HTML

- You can serve up any kind of binary file. Often have associated MIME-type like with e-mail
- Browser also often has built in support
- GIF (trouble due to patents), PNG. SVG?
- MP3 music? Movies?
- Plugins. Flash? Java? PDF?



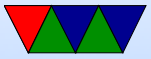
Web-servers

- famously netcraft had a list (meme netcraft reports BSD is dying)
- NCSA was first popular one
- Apache ("a patchy" version of NCSA) took over
- Microsoft IIS
- Other companies like Sun/Netscape/SGI
- nginx ("engine-x")
  Designed to be faster than apache (Apache has lots of RAM overhead)



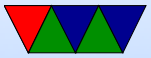
Solve c10k problem (having 10k active socket connections at once)

- lighthttpd ("lightly")

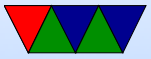


simple web server

- Listen on port 80
- Accept a TCP connection
- Get name of file requested
- Read file from disk
- Return to client
- Release TCP connection
- How do we make this faster?
  - Cache things so not limited by disk
    (also cache in browser so not limited by network)



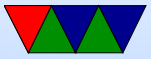
- Make server multithreaded



URLs

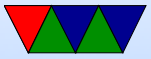
- URI (uniform resource identifier)
- URL (uniform resource locator) subset of URI, includes info on how to find the resource (protocol and server)

- URN (uniform resource name) asks for a document but from anywhere. I.e. give it something like an ISBN and returns the book
- `scheme:[//[user:password@]host[:port]][/]path[?query][#fragment]`
  `: / ? # [ ] @ reserved, must encode if use %3f`
  `query key1=value1&key2=value2 or key1=value1;key2=value2`
- protocol: http, ftp, file, news, gopher, mailto, telnet



http

- HyperText Transfer Protocol
  RFC 2068 (1997), RFC 2616 (1999), RFC 7230 (2016)
- Make ASCII request, get a MIME-like response
- HTTP 1.0, single request was set and single response
  HTTP 1.1 supports persistent connections, allowing multiple requests to happen with one TCP connection (lowering overhead)
- Commands
  - GET filename HTTP/1.1



get file

- HEAD

  get header (can check timestamp. why? see if cache up to date)

- PUT

  send a file

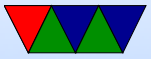
- POST

  append to a file

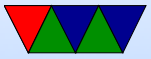
- DELETE

  remove file (not used much)

- TRACE



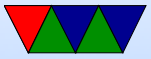
debugging
  - CONNECT, OPTIONS
- Responses are three digit status codes
  - 1xx – informational – not used much
  - 2xx – Success – 200 = page is OK
  - 3xx – Redirect – 303 = page moved
  - 4xx – Client Error – 403 = forbidden, 404 = not found
  - 5xx – Server Error – 500 = internal, 503 = try again
- Additional request headers. A lot are possible, included after the GET.
  - User-Agent (browser info). Can you lie? Can you leak



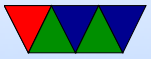
info?

- Accept-*: type of documents can accept, compression, character set
- Host: server you are requesting
  Can configure browser to open up helper util for this (for example, run Office if it's a word file)
- Authorization: if you need special permissions/login
- Cookie: deals with cookies
  Statelessness – how do you remember setting, logins, shopping cart, etc. "cookies". Expire. Can be misused.

- Additional response headers.

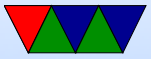


- Content-Encoding,Language,Length,Type
- Last-Modified: helps with caching
- Location: used when redirecting
- Accept-Ranges: partial downloads (downloading a large file, interrupted, can restart where left off)



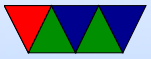
Do you need a browser?

```
telnet www.maine.edu 80
GET / HTTP/1.1
Host: www.maine.edu
(enter)(enter)
control-]
close
```



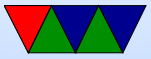
How simple cane a server be?

- My Apple II webserver project `http://www.deater.net/weave/vmwprod/apple2_eth/`

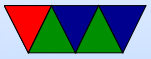


HTTP/2

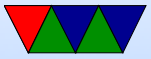
- 2015. RFC 7540
- Google push through, extension of their SPDY (speedy) Microsoft and Facebook giving feedback
- Why does google care about (relatively) small increases in web performance?
- Leaves a lot of high level things the same. Negotiate what level to use.
- Decrease latency of rendering web pages:
  - compress headers



- Server can push data the browser didn't request yet but it knows it will need (like images, etc)
- pipeline requests
  Send multiple requests without waiting for response
  good on high-latency links (FIFO on 1.1, new makes it asynchronous)
- multiplex multiple requests over one TCP connection
- head-of-line blocking problem?
  line of packets held up by processing of first
  FIFO first requests waits until done until next, can't run in parallel

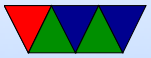


- Page load time 10-50% faster
- While can use w/o encryption, most browsers say will only do with encryption
- Criticism: was rushed through. Is way complex. Does own flow control (has own TCP inside of TCP) Re-implements transport layer at application layer



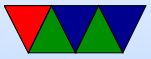
What if Server Overloaded?

- Slashdot effect (modern: HackerNews?)
- caching/proxy – squid
- Content Delivery Network – akami
- Server farms



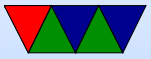
Security

- SSL – Secure Socket Layer
- Replaced by TLS (Transport Layer Security)
- Port 443 for https
- Public key encryption.



Setting Up a Web-server

- Apache



Web Seach

- Web-bots index the web. robots.txt file
- Altavista, Hotbot, Excite, Inktomi, etc.
- Curated search like Yahoo (people organize links rather than automatically search)
- Google (1996 some machine in Stanford, 1997-1998)
- MSN search 1999, rebranded Microsoft Bing 2009