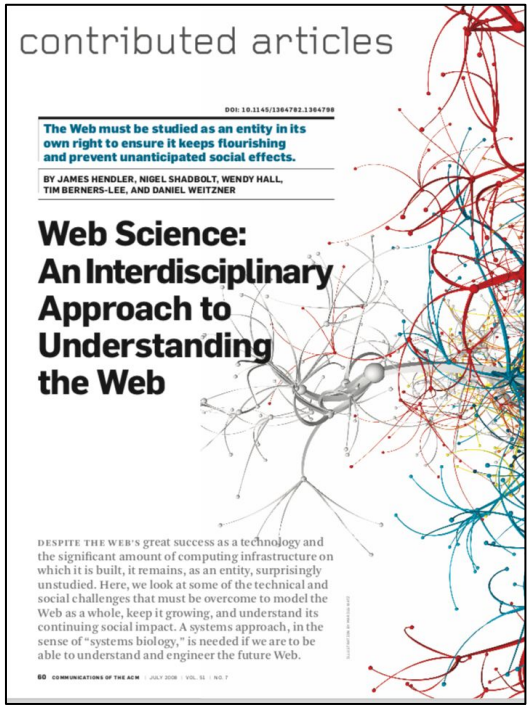
Web Science:
Web Science and Web Architecture
(Part 1 - Intro to Web Science)

CS 432/532

Old Dominion University

Permission has been granted to use these slides from Frank McCown, Michael L. Nelson, Alexander Nwala, Michele C. Weigle

Assigned Reading/Viewing



[Web Science: An Interdisciplinary Approach to Understanding the Web | July 2008]

[What is Web Science?], YouTube

"On the 10th anniversary of this emerging interdisciplinary research field, Professors Leslie Carr, Dave De Roure, Dame Wendy Hall, Sir Nigel Shadbolt, Noshir Contractor, Ted Nelson, Manfred Hauswirth, Susan Halford, with Dr Pete Burnap and Switch Concepts CEO, Tom Barnett, give their views on the nature of Web Science." Dec 5, 2016

What is Web Science?

Web Science is the interdisciplinary study of the Web as an entity and phenomenon. It includes studies of the Web's properties, protocols, algorithms, and societal effects.

Web Science: An Interdisciplinary Approach to Understanding the Web | July 2008

Background

Web Science initiative launched in Nov 2006 by University of Southampton and MIT

Sir Nigel Shadbolt

Sir Tim Berners-Lee

Dame Wendy Hall

James Hendler

Daniel Weitzer

Images from [Web Science Trust Board]

Web Science is Not...

- Web page design (HTML, CSS)
- JavaScript programming
- How to use the Internet
- Computer networking

The Web Itself is Worthy of Study

- "The Web is the most used and one of the most transformative applications in the history of computing, even of human communications."
- "There is significant interplay among the social interactions enabled by the Web's design, the scalable and open applications development mandated to support them, and the architectural and data requirements of these large-scale applications."

Web Science: An Interdisciplinary Approach to Understanding the Web | July 2008

Web Science is Engineering, Hard Science, *and* Social Science

- "The Web is an infrastructure of artificial languages and protocols; it is a piece of *engineering*."
- "However, it is the *interaction of human beings* creating, linking, and consuming information that generates the Web's behavior as emergent properties at the macro scale."

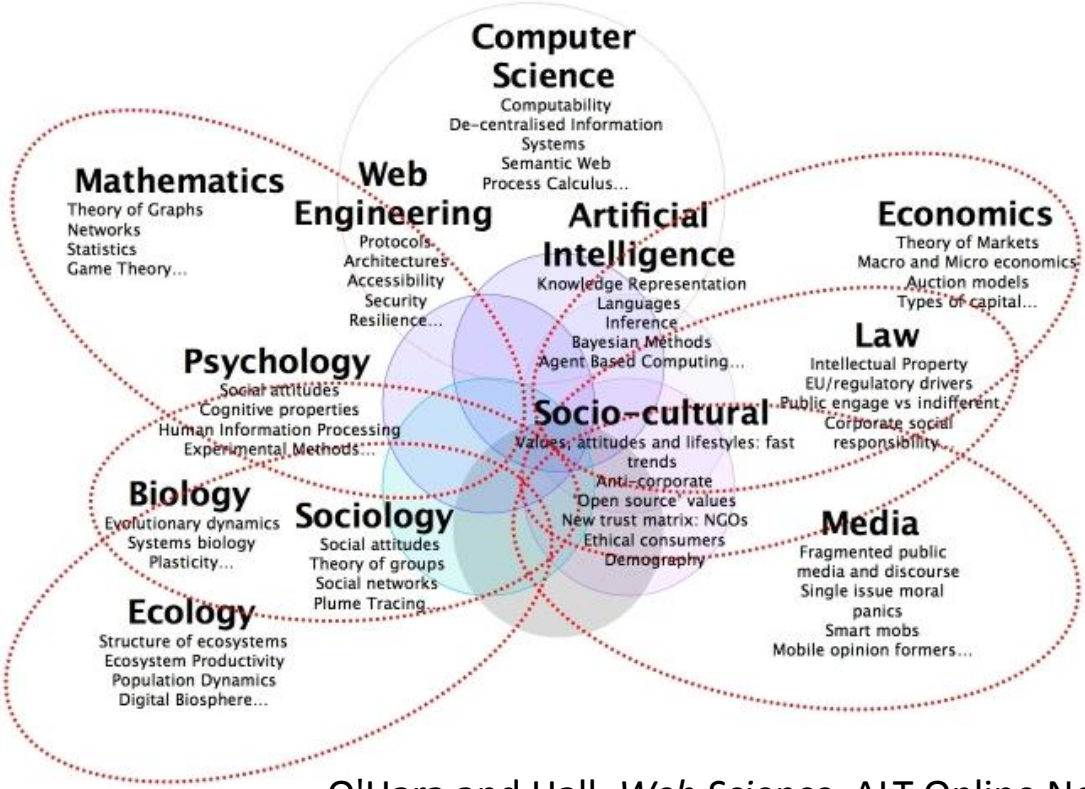
Web Science: An Interdisciplinary Approach to Understanding the Web | July 2008

Web Science is Interdisciplinary

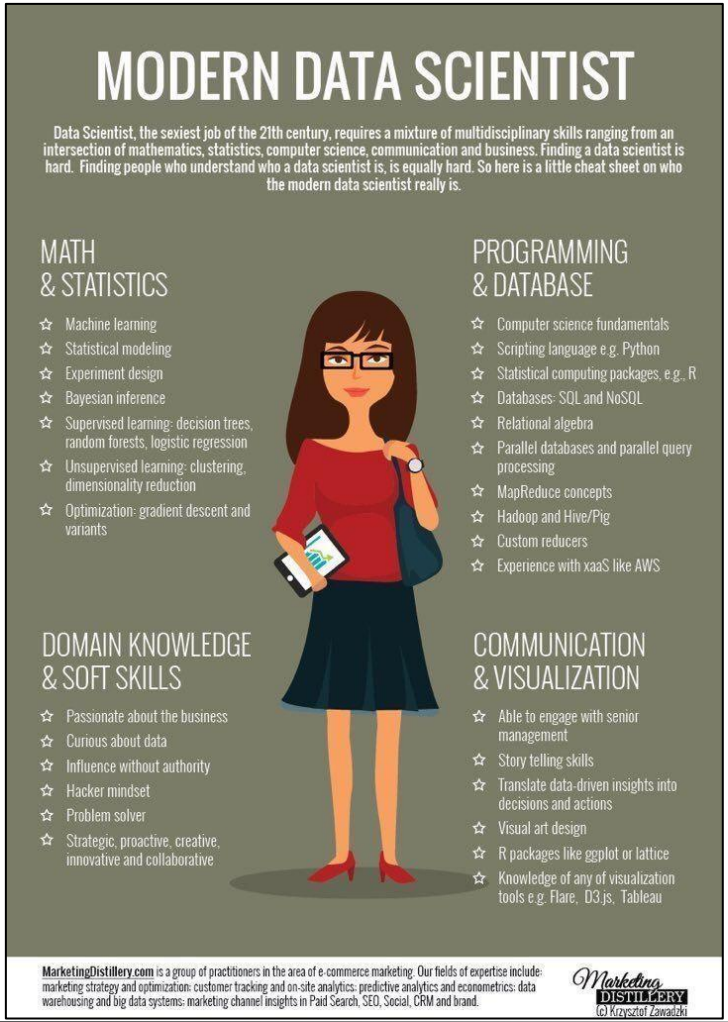
"Given the breadth of the Web and its inherently multi-user (social) nature, its **science is necessarily interdisciplinary**, involving at least mathematics, CS, artificial intelligence, sociology, psychology, biology, and economics. We invite computer scientists to **expand the discipline** by addressing the challenges following from the widespread adoption of the Web and its profound influence on social structures, political systems, commercial organizations, and educational institutions."

Web Science: An Interdisciplinary Approach to Understanding the Web | July 2008

Web Science is Interdisciplinary



O'Hara and Hall, *Web Science*, ALT Online Newsletter, May 6, 2008



Web science *is* data science.

If you want a job as a "data scientist", this class will prepare you.

Image source: [Dataiku on Twitter: "The Definitive Q&A for Aspiring #DataScientists… "]

Some questions of study:

- How is the Web **structured**? What is its **size**?
- How can **unstructured data** mined from the Web be combined in meaningful ways?
- How does **information/misinformation** spread on the Web? How can we discover its **origin**?
- How can the Web us
  intelligence of its us
- How can **trust** be m
- How can **privacy** be
- What do events gathered from **online social networks** tell us about the human condition?
- Has the Web changed how humans **think**?

Why is this important?

Huge implications for web search!

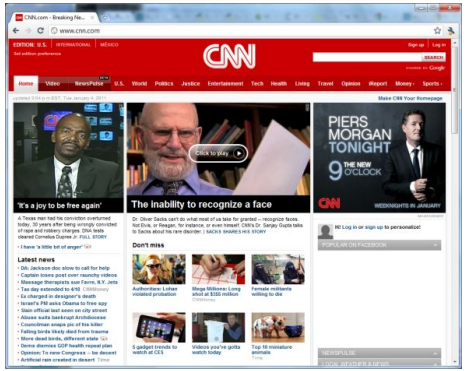
Web Science:
Web Science and Web Architecture
(Part 2 - Structure and Size of the Web)

CS 432/532

Old Dominion University

Permission has been granted to use these slides from Frank McCown, Michael L. Nelson, Alexander Nwala, Michele C. Weigle

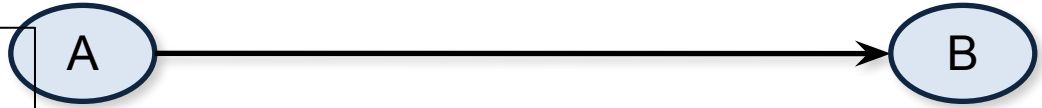
How is the Web structured?



link

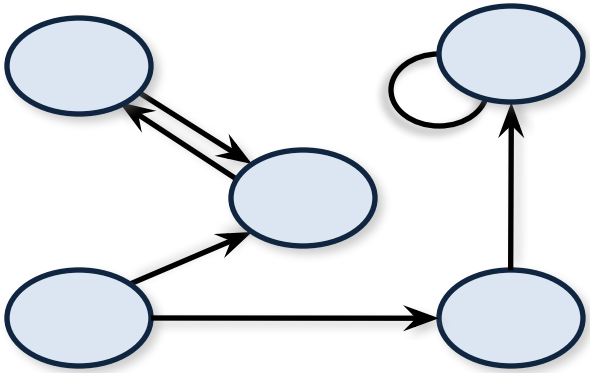
In-link, in-degree
(A = 0, B = 1)

Out-link, out-degree
(A = 1, B = 0)



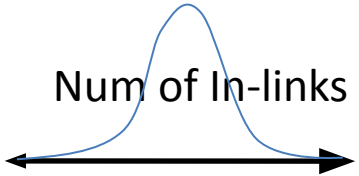
Graph Theory: Pages are nodes & links are directed edges

Web Graph

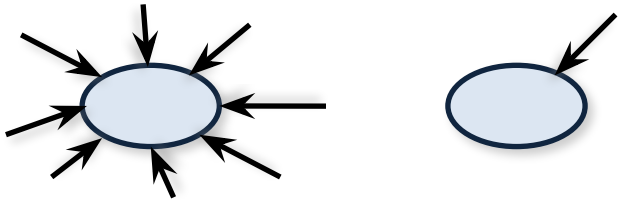


Random Graph

Normal/Gaussian Distribution

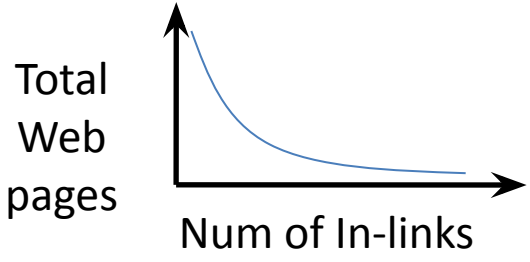


(decrease) Total Web pages (increase)



Typical Web Graph

Power-law Distribution



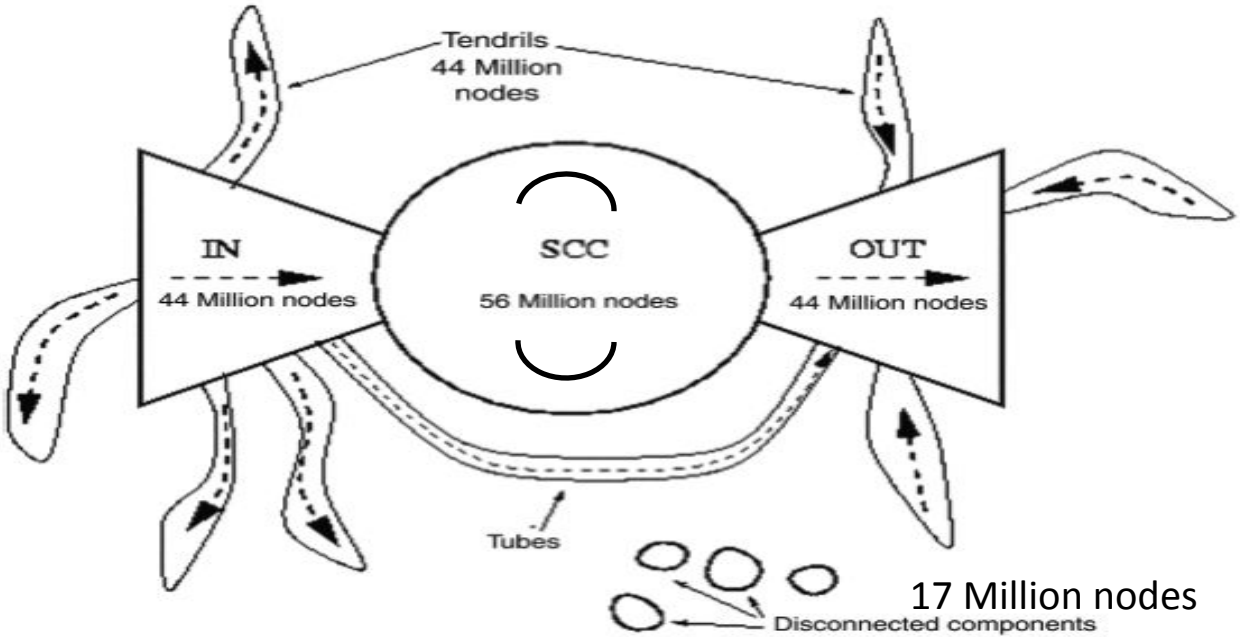
Small World Network

- Six degrees of separation
- Most pages are not neighbors but most pages can be reached from others by a small number of hops
- Many hubs - pages with many in-links
- Robust for random node deletions
- Other examples: road maps, networks of brain neurons, voter networks, and social networks

Six Degrees of Kevin Bacon

[The Oracle of Bacon]

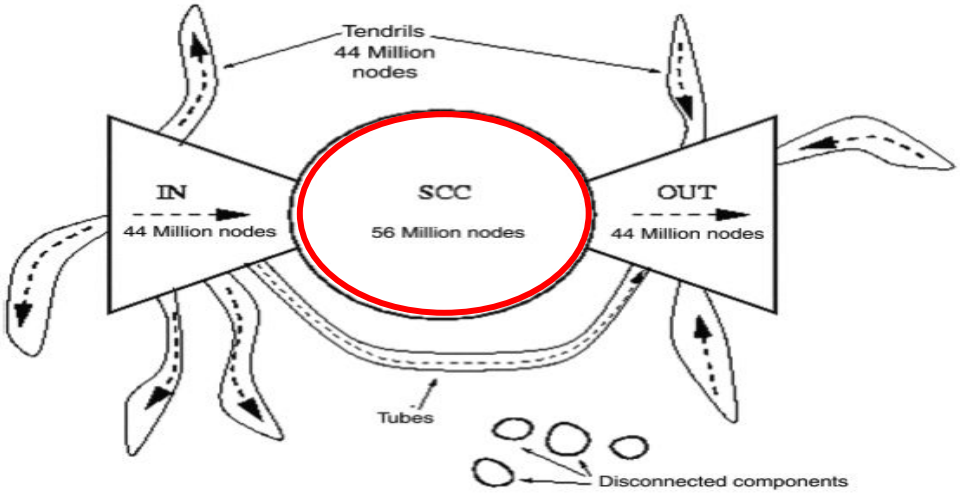
Bow-Tie Structure of the Web



Examined a large web graph (200M pages, 1.5B links)

Broder et al., Graph Structure of the Web, 2000

Bow-Tie Structure of the Web

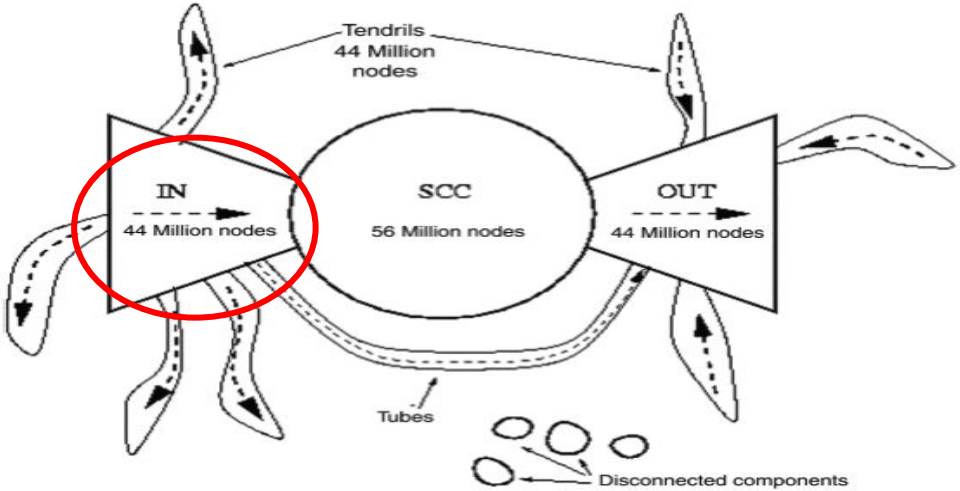


SCC: Strongly-connected component - all nodes here can reach one another along directed links

Pages with *in-links* from IN or SCC and *out-links* to OUT or SCC.

Broder et al., Graph Structure of the Web, 2000

Bow-Tie Structure of the Web



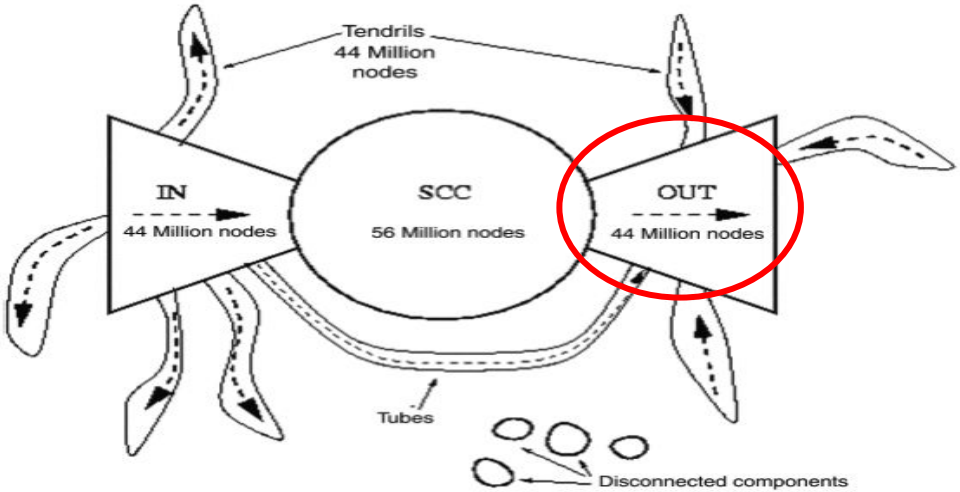
IN: Pages that can reach the SCC but cannot be reached from it.

Pages with no *in-links,* or with *in-links* from IN pages and *out-links* to pages in IN, SCC, Tendrils, or Tubes.

Example: New sites that people have not yet discovered and linked to

Broder et al., Graph Structure of the Web, 2000

Bow-Tie Structure of the Web



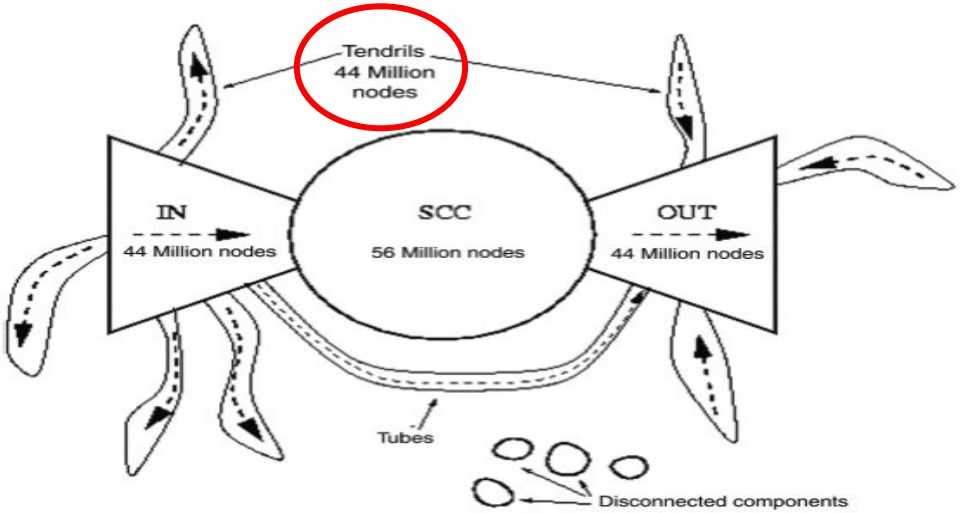
OUT: Pages that are accessible from the SCC, but do not link back to it.

Pages with no *out-links,* or with *out-links* to other pages in OUT and all *in-links* come from OUT, SCC, Tendrils, or Tubes.

Example: Corporate websites that contain only internal links

Broder et al., Graph Structure of the Web, 2000

Bow-Tie Structure of the Web

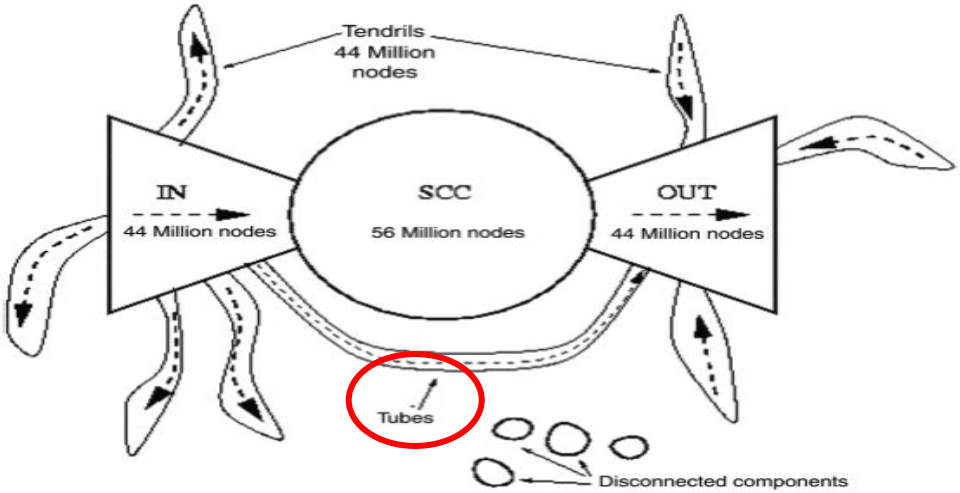


Tendrils: Pages that cannot reach the SCC and cannot be reached from the SCC.

Pages that can only be reached from IN, or can only reach OUT.

Broder et al., Graph Structure of the Web, 2000

Bow-Tie Structure of the Web

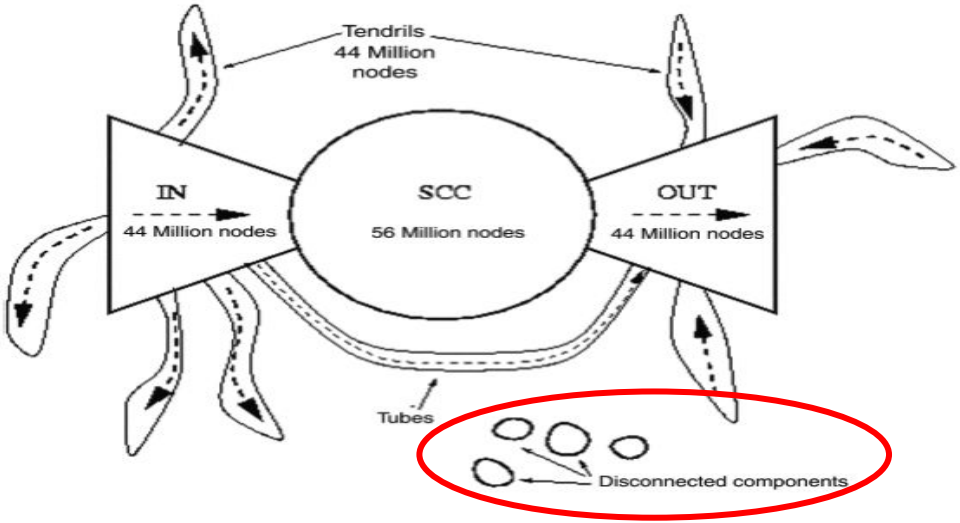


Tubes: Connects a TENDRIL hanging off from IN to a TENDRIL leading into OUT (a passage from a portion of IN to a portion of OUT without touching SCC)

Pages that have *in-links* from IN or other pages in Tubes and *out-links* to pages in Tubes or OUT.

Broder et al., Graph Structure of the Web, 2000

Bow-Tie Structure of the Web



Disconnected: Pages that have no *in-links* from any other components and no *out-links* to other components. These pages may be linked to each other.

Broder et al., Graph Structure of the Web, 2000

SCC

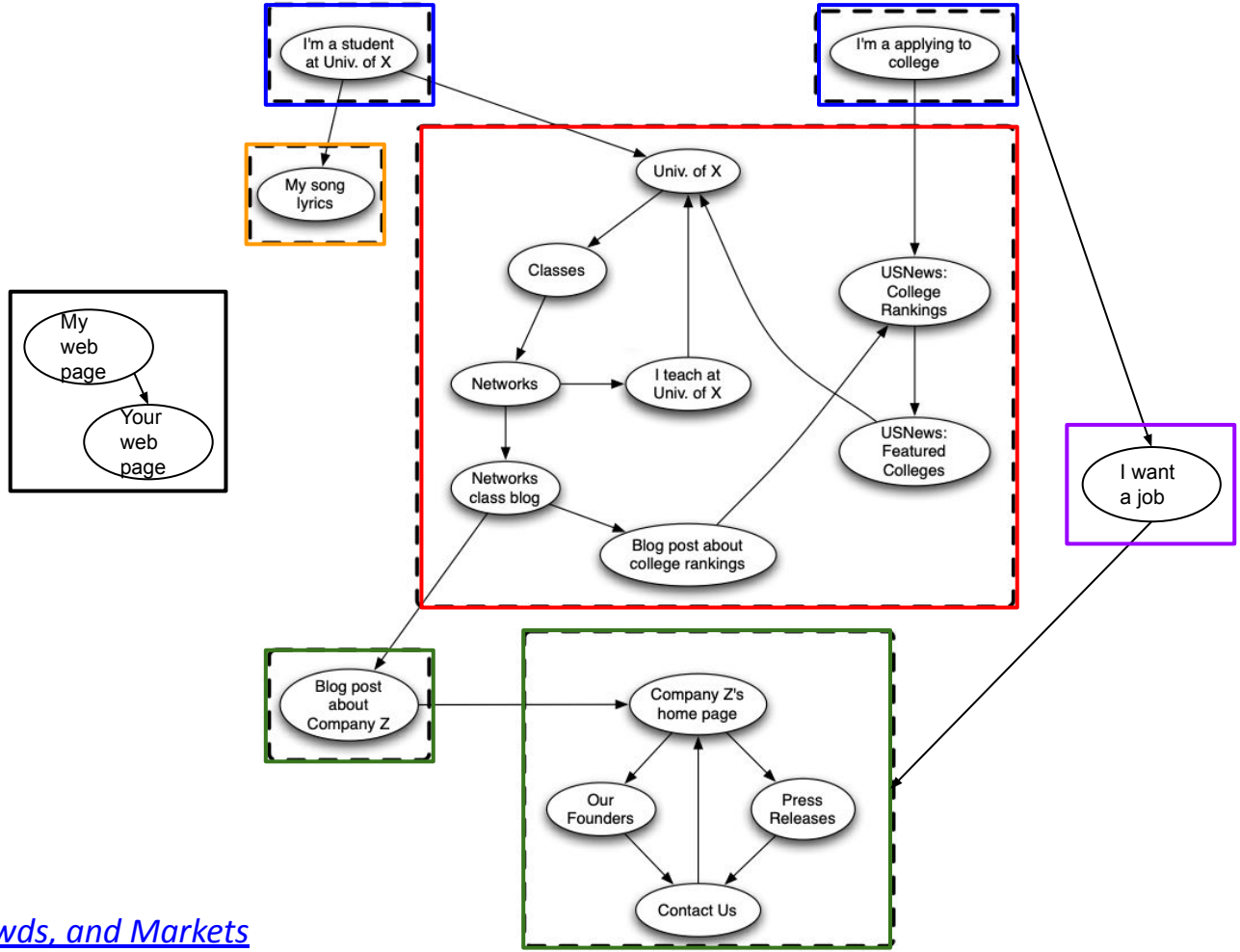
IN

OUT

tendril

tube

disconnected



Based on Fig 13.6 from *Networks, Crowds, and Markets*

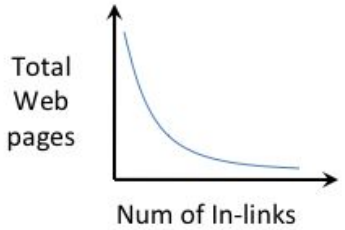
Bow-Tie Structure

- 75% of pages do not have a direct path from one page to another
- Avg distance is **16 clicks** when directed path exists and **6 clicks** when undirected path exists
- Diameter of SCC is at least 28 (max shortest distance between any two nodes)
- Diameter of entire Web is at least 500 (most distant node in IN to OUT)

Broder et al., Graph Structure of the Web, 2000

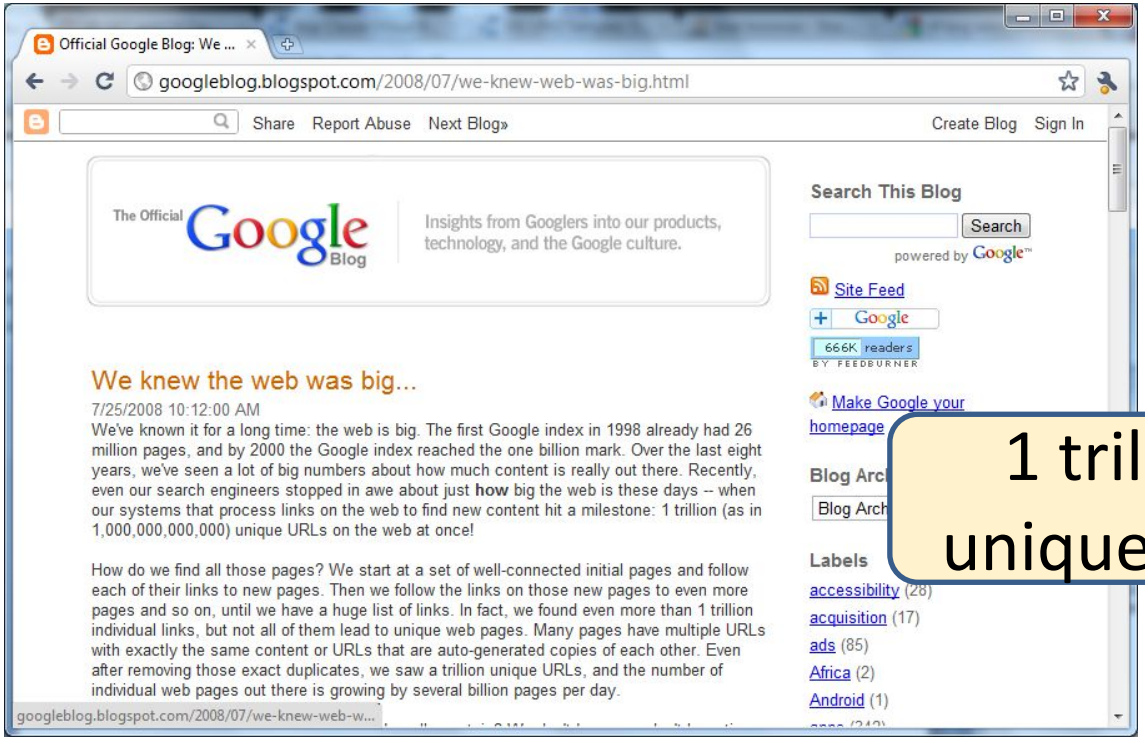
Web Structure's Implications

- If we want to discover every web page on the Web, it's impossible since there are many pages that aren't linked to
  - finding popular pages is easy, but finding pages with few in-links (the long tail) is more difficult



- How do we know when new pages are added to the Web or removed?
- Incoming links could tell us something about the "importance" of a page when searching the Web for information (e.g., PageRank)
  - link structure of the Web can be artificially manipulated

How large is the Web?



[Official Google Blog: We knew the web was big...]

How did Google discover all these URLs?

By crawling the web

Web Crawler

Web crawlers are used to fetch a page, place all the page's links in a queue, and continue the process for each URL in the queue

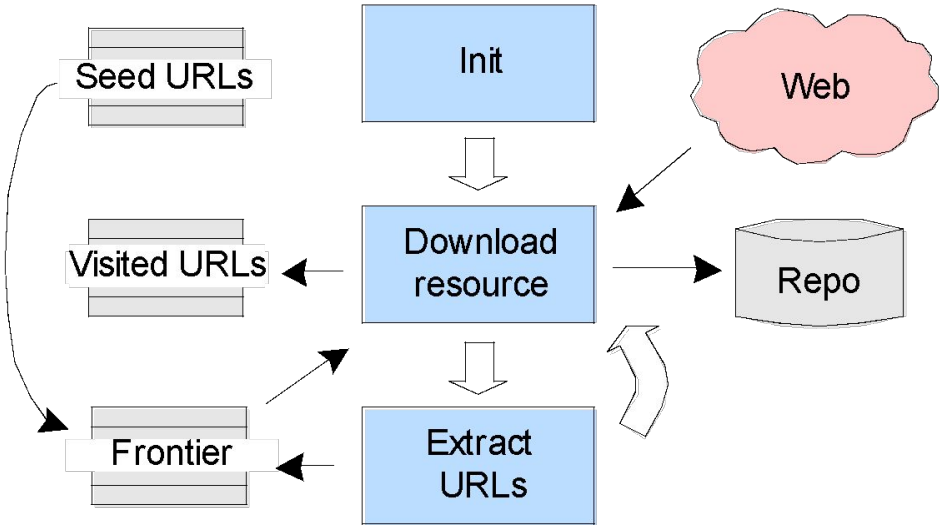


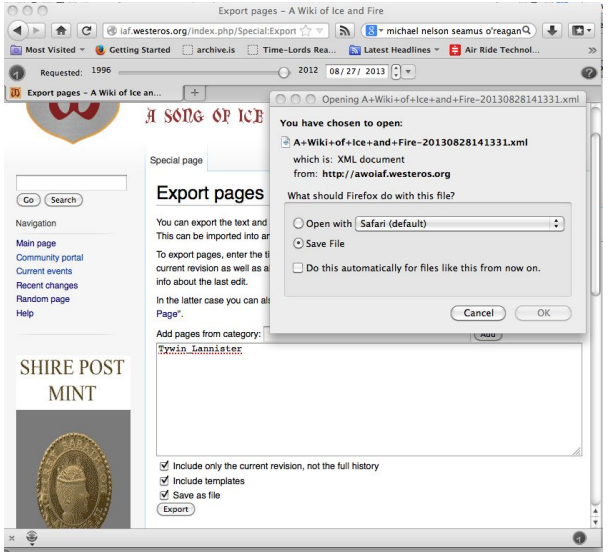
Figure from McCown, *Lazy Preservation: Reconstructing Websites from the Web Infrastructure*, Dissertation, 2007.
See also: Web crawler (Wikipedia)

Problems with Web Crawling

- Slow because crawlers limit how frequently they make requests to the same server (*politeness policy*)
- Many pages are disconnected from the SCC, password-protected, or protected by robots.txt
- There are an **infinite number** of pages (e.g., calendar) so crawlers limit how deeply they crawl
- Web pages are continually being added and removed
- **Deep web**: Many pages are only accessible behind a web form (e.g., US patent database). Deep web is magnitudes larger than surface web, and 2006 study[1] shows only 1/3 is indexed by big three search engines

[1]He et al., Accessing the deep web, *CACM* 2007

Deep Web != Dynamic, Queries, or Personalized



not deep web:

http://oracleofbacon.org/movielinks.php?game=0&a=Kevin+Bacon&b=Seamus+O%27Regan&use_using=1&u0=on&u1=on&use_genres=1&g0=on&g4=on&g8=on&g16=on&g20=on&g24=on&g1=on&g5=on&g9=on&g13=on&g17=on&g21=on&g25=on&g2=on&g6=on&g10=on&g14=on&g18=on&g22=on&g26=on&g3=on&g7=on&g11=on&g15=on&g23=on&g27=on

deep web:

http://awoiaf.westeros.org/index.php/Special:Export
(or more accurately, the 1000s of XML files available from this same URI are in the deep web)

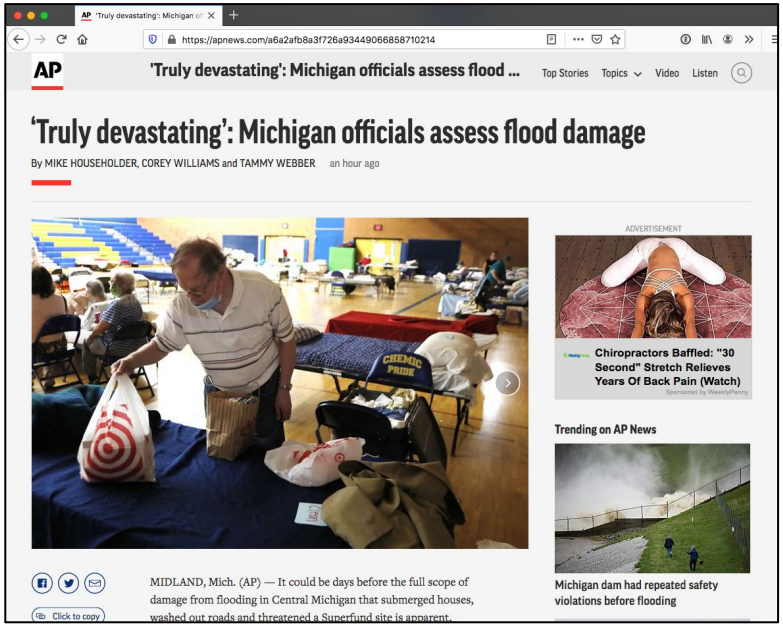
What Counts?

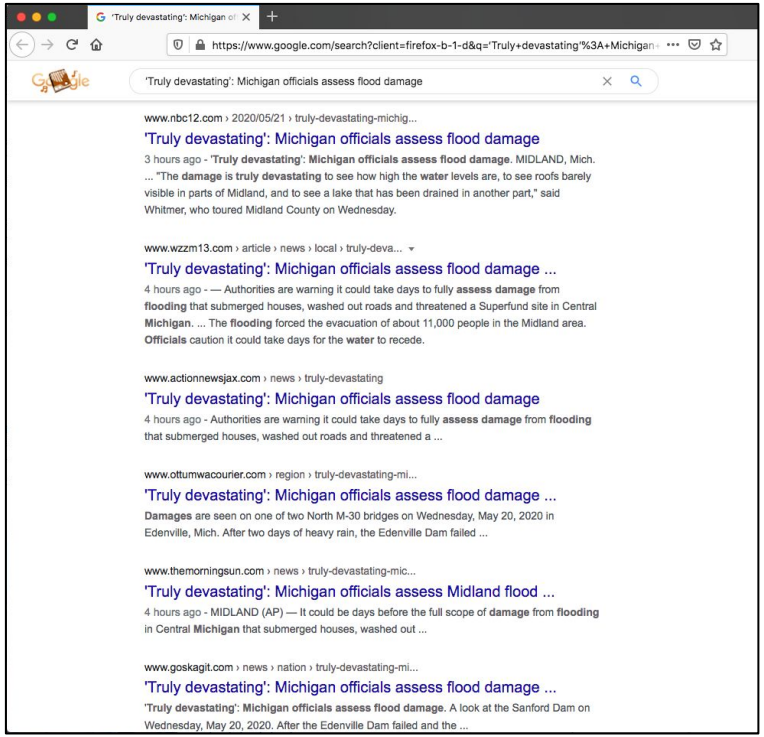
- Many duplicate pages (30% of web pages are duplicates or near-duplicates[1])
  - How do we efficiently compare across a large corpus?
- Some pages change every time they are requested
  - How can we automatically determine what is an insignificant difference?
- Many spammy pages (14% of web pages[2])
  - How can we detect these?

[1]Fetterly et al., On the evolution of clusters of near-duplicate web pages, *J of Web Eng*, 2004

[2]Ntoulas et al., Detecting spam web pages through content analysis, WWW 2006

Duplicates & Near Duplicates





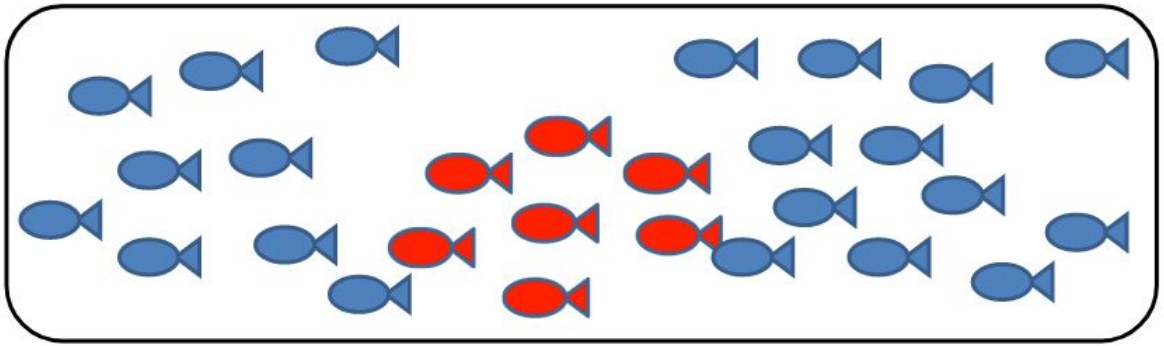
Some Observations

- Crawling a significant amount of the Web is hard
- Different search engines have different pages indexed, but they don't share these differences with each other (company secret)
- So if we wanted to estimate the Web's size but don't want to try to crawl the Web ourselves, could we use the search engines themselves to estimate the Web's size?

  (note: working with the web == working with stats)

Capture-Recapture Method

- Statistical method used to estimate population size (originally fish and wildlife populations)

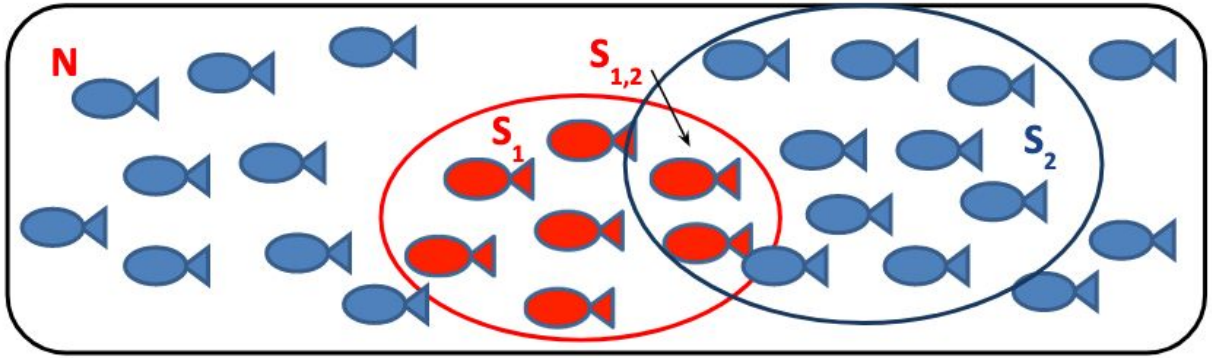


Lincoln, Frederick C. (May 1930). Calculating Waterfowl Abundance on the Basis of Banding Returns. Circular. **118**. Washington, DC: United States Department of Agriculture.

Petersen, C. G. J. (1896). "The Yearly Immigration of Young Plaice Into the Limfjord From the German Sea", Report of the Danish Biological Station (1895), 6, 5–84.

Capture-Recapture Method Example

- How many fish are in the lake?
  - Catch S_1 fish from the lake, tag them, and return them to the lake
  - Then catch and put back S_2 fish, noting which were already tagged (now, $S_{1,2}$)
  - $S_1/N = S_{1,2}/S_2$ so population $N = S_1 \times S_2/S_{1,2}$



Estimate Web Population

- Lawrence and Giles[1] used capture-recapture method to estimate web page population
  - Submitted 575 queries to sets of 2 search engines
  - S1 = All pages returned by SE1
  - S2 = All pages returned by SE2
  - S1,2 = All pages returned by both SE1 and SE2
  - Size of indexable Web (N) = $S_1 \times S_2/S_{1,2}$
- 1998 - estimated size of indexable Web = 320 M pages
- July 2020 - estimates of lower bound = 5.5 B pages[2]

[1]Lawrence & Giles, Searching the World Wide Web, *Science*, 1998

[2]The size of the World Wide Web (The Internet)

This is just a sample of Web Science that we will be examining from a *computing perspective*.

Web Science:
Web Science and Web Architecture
(Part 3 - Web Architecture and HTTP)

CS 432/532

Old Dominion University

Permission has been granted to use these slides from Frank McCown, Michael L. Nelson, Alexander Nwala, Michele C. Weigle

Internet != Web

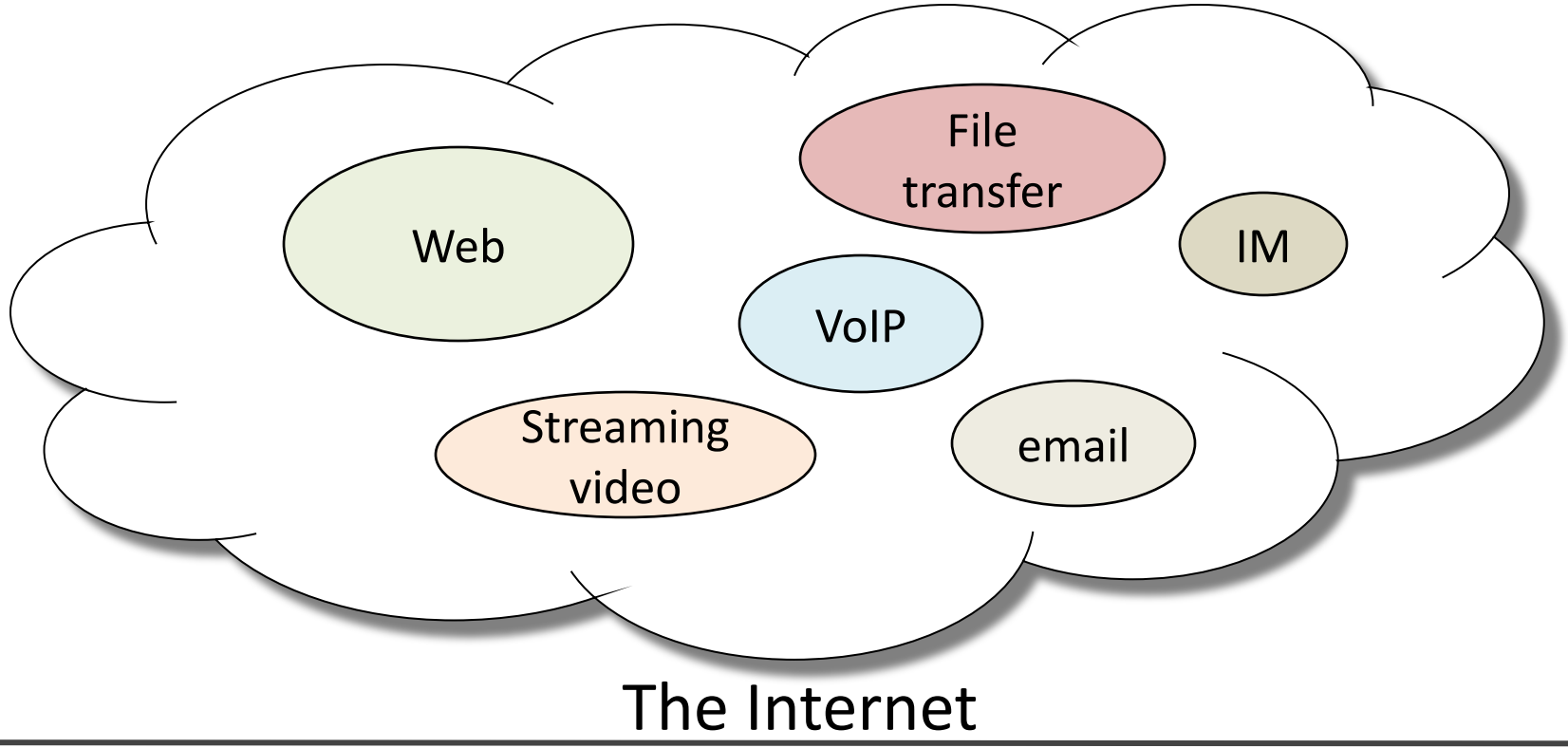
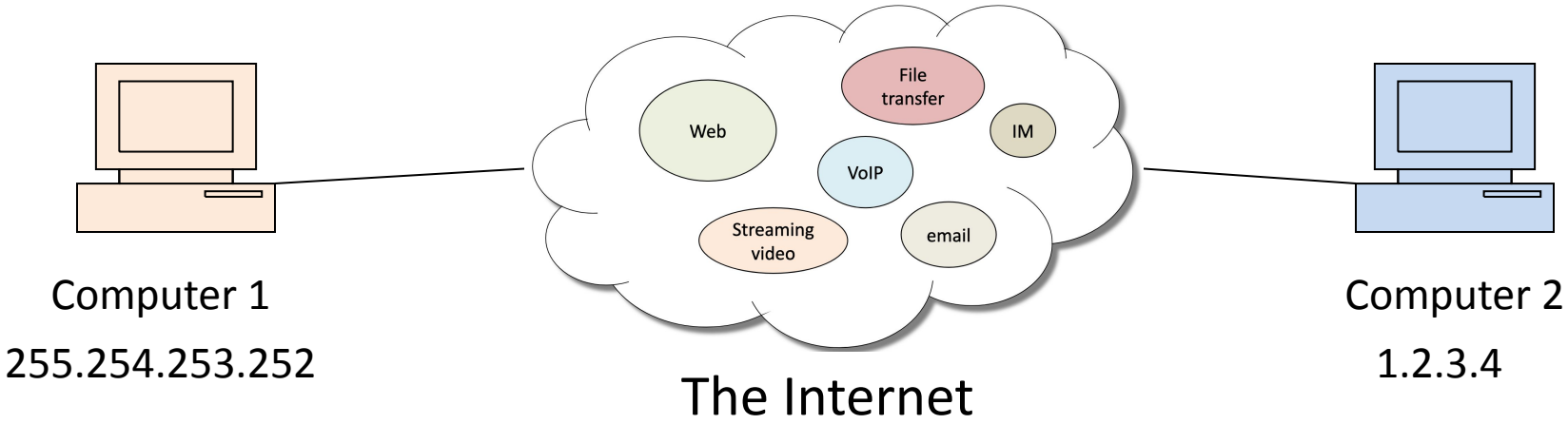


Image from: Internet Hall of Fame News Highlights

"The **Internet** is a global system of interconnected computer networks that use the standard Internet Protocol Suite (TCP/IP) to serve billions of users worldwide."

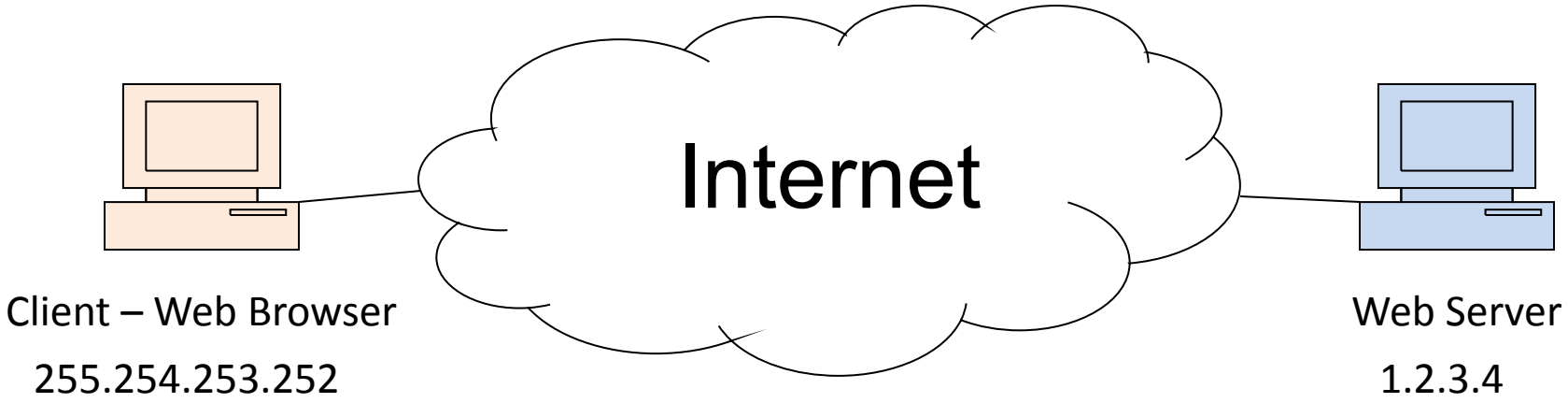
Internet (Wikipedia)



Internet Protocol Suite

- **Internet Protocol (IP)**: directs packets to a specific computer using an IP address
- **Transmission Control Protocol (TCP)**: directs packets to a specific application on a computer using a port number.
  - Common port numbers:
    - 22 – ssh
    - 23 – telnet
    - 25 – email
    - 80 – Web (HTTP, non-secure)
    - 443 - Web (HTTPS)

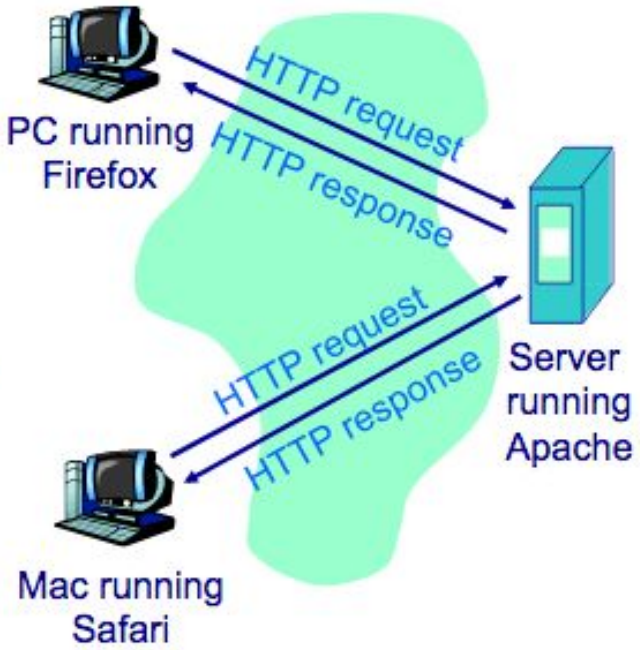
Overview of the Web



World Wide Web: The system of interlinked hypertext documents accessed over the Internet using the HTTP protocol.

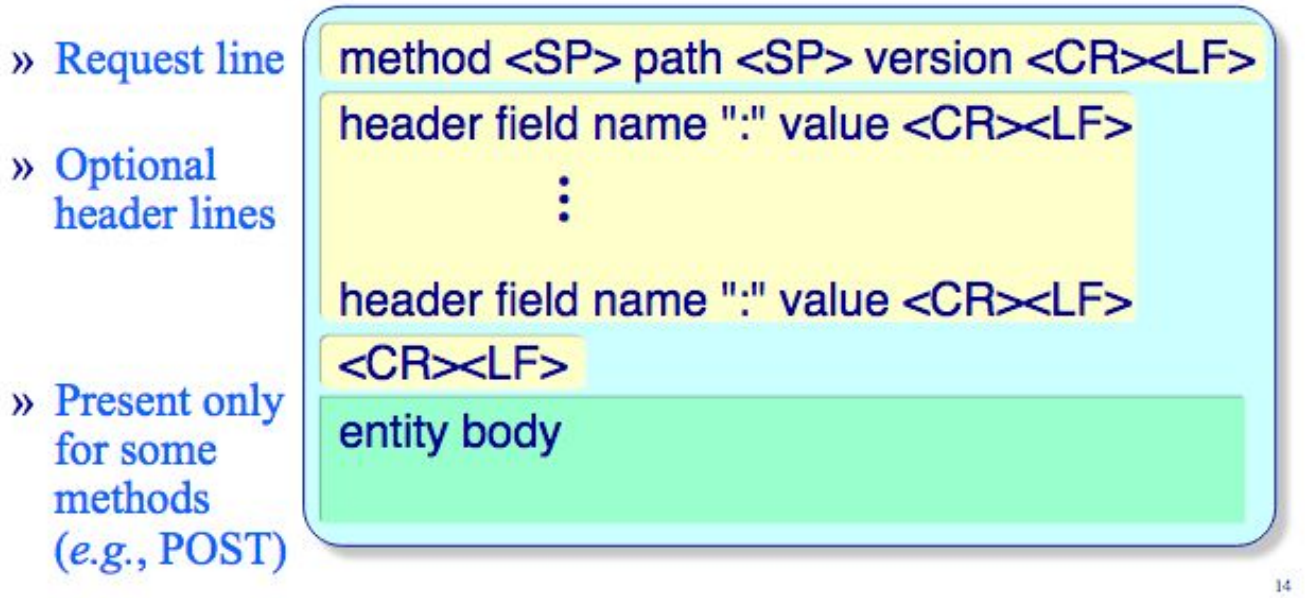
Hypertext Transfer Protocol (HTTP)

HTTP is the set of rules that govern communication between web browsers and web servers.



From CS 455/555 Course Notes

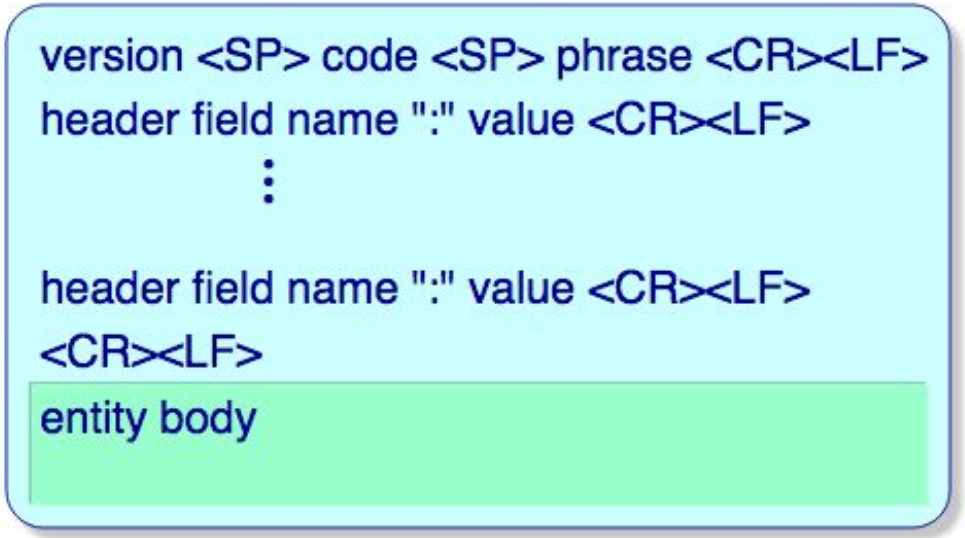
HTTP Request Format



From [CS 455/555 Course Notes]

HTTP Response Format

- Status line
- Optional header lines
- Requested object, error message message, etc.



From CS 455/555 Course Notes

Example HTTP Request, Response

Requesting http://www.harding.edu/comp/

Client Request

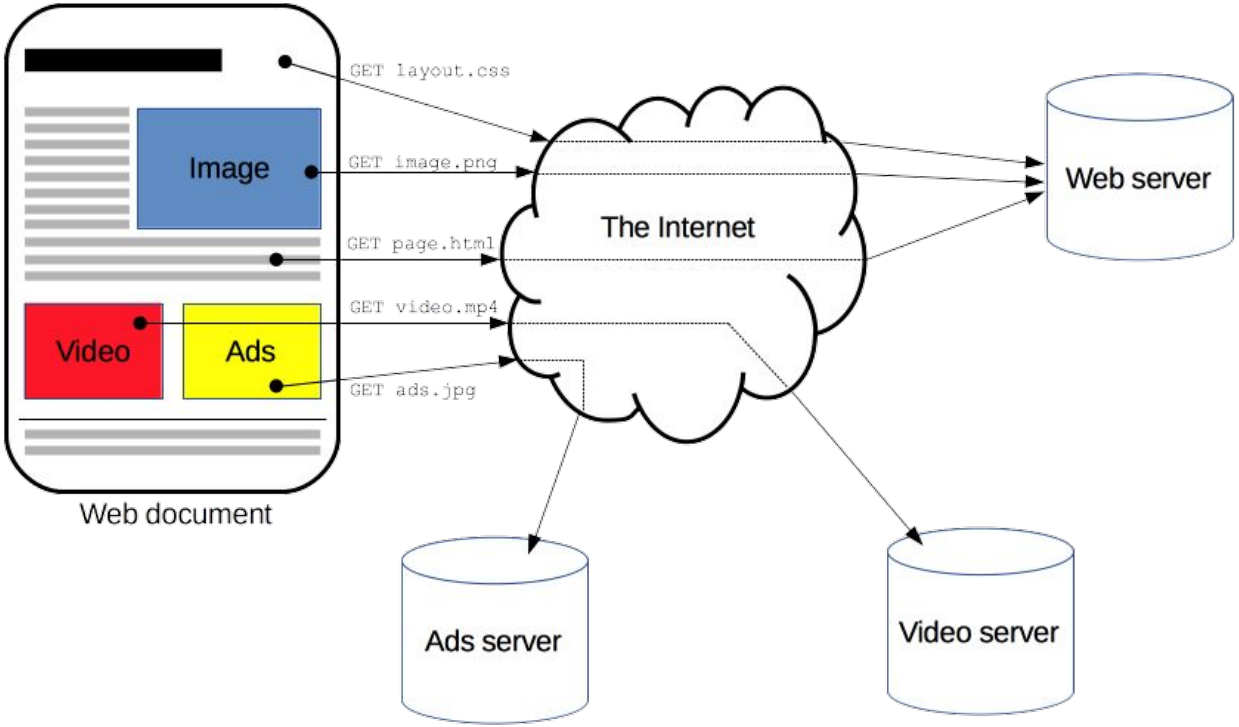
```
GET /comp/ HTTP/1.1
Host: www.harding.edu
```

Server Response

```
HTTP/1.1 200 OK
Content-Length: 6018
Content-Type: text/html
Content-Location: http://www.harding.edu/comp/
Last-Modified: Mon, 05 Jul 2010 18:49:40 GMT
Server: Microsoft-IIS/6.0

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd"> <html> <head> <title>Harding
University - Computer Science</title>
```

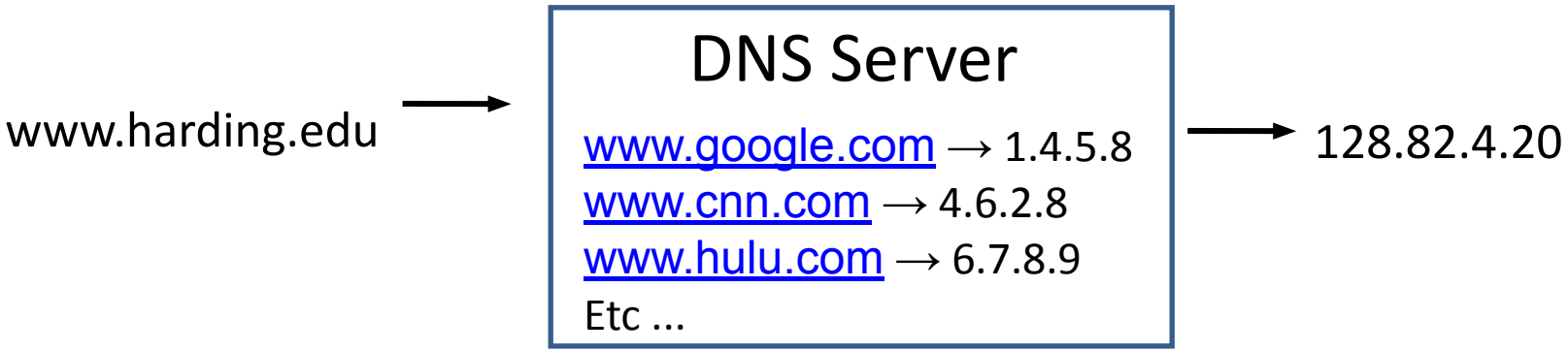
Building Up a Webpage



Learn more about HTTP: HTTP (Mozilla Developer Docs)

Domain Name System (DNS)

DNS is a hierarchical look-up service that converts a given hostname into its equivalent IP address



- DNS servers contact parent servers for missing entries
- Authoritative name servers are responsible for specific domains

Domain Name System (Wikipedia)

Hierarchical DNS

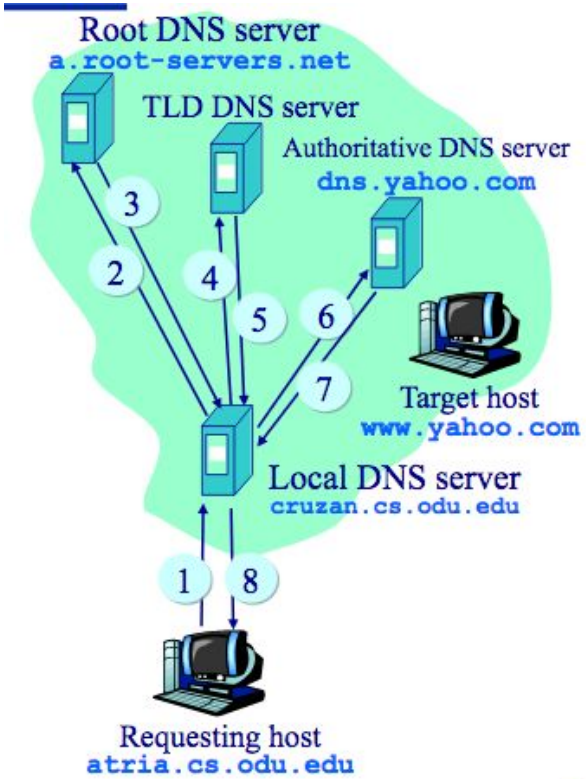
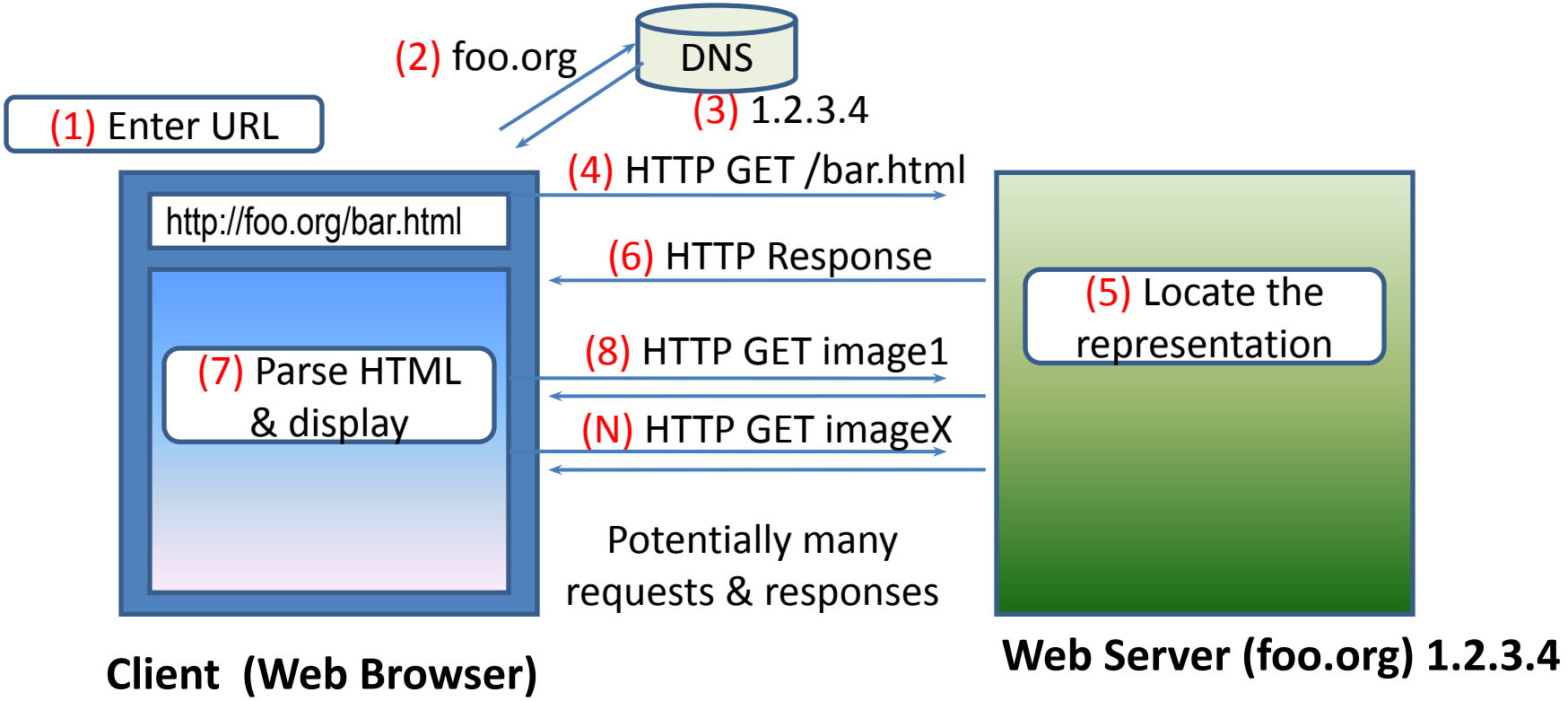


Image from: CS 455/555 Course Notes

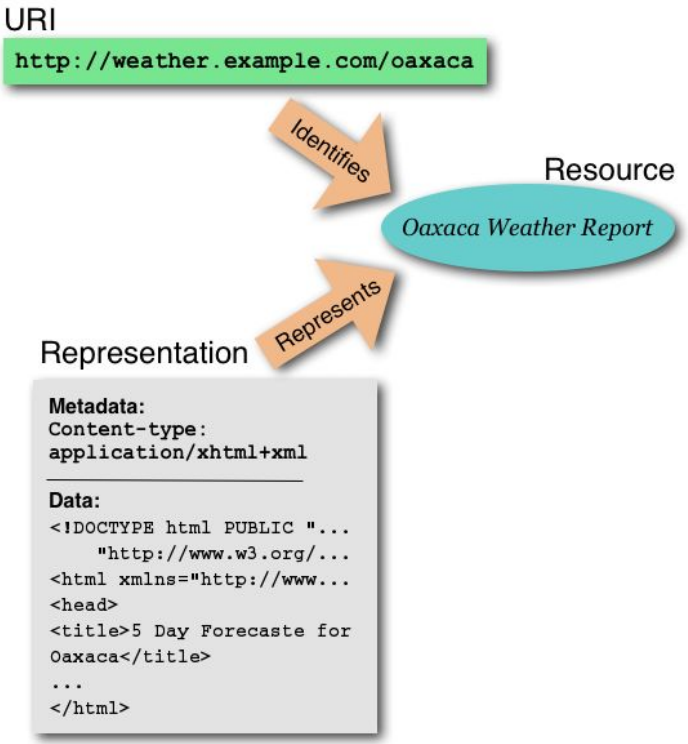
Example: Web Page Request



More Formal Definitions

- HTTP defined by Request for Comments (RFCs) 1945, 2068, and 2616
  - technically recently replaced by RFCs 7230—7235, see: mnot's blog: RFC2616 is Dead
- Other RFCs for defining URLs (1736, 1738), URIs (1630, 2396), etc.
- Web architecture defined in W3C's *The Architecture of the World Wide Web, Volume One*

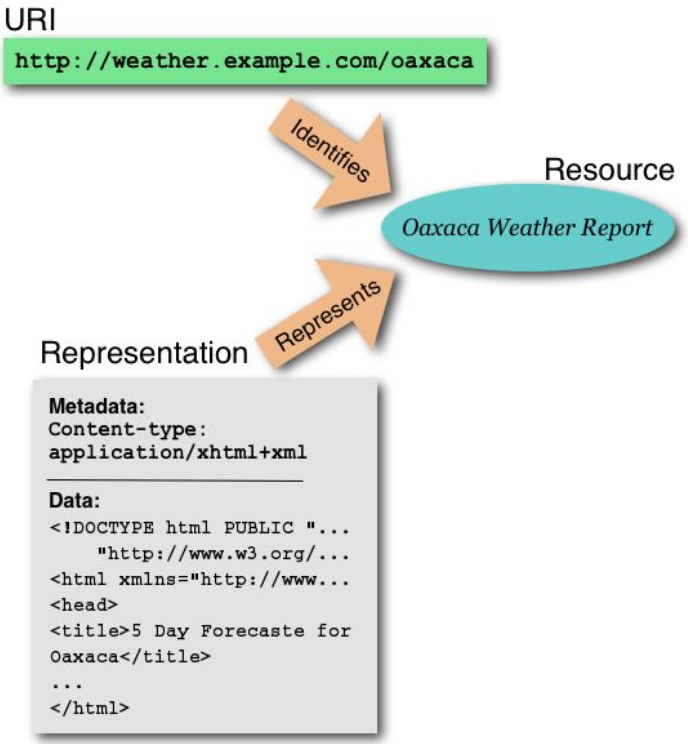
Web Definition



"The ***World Wide Web*** (***WWW***, or simply ***Web***) is an information space in which the items of interest, referred to as resources, are identified by global identifiers called Uniform Resource Identifiers (***URI***)."

Architecture of the World Wide Web, Volume One

URIs, Resources, and Representations



- URIs *identify* Resources
- Representations *represent* Resources
- When URIs are dereferenced, they return representations (not resources)
- Different representations may be returned for the same URI (e.g., English vs. French version)

Architecture of the World Wide Web, Volume One

Remember Three Things

URIs

http://www.cs.odu.edu/~mln/

Identify

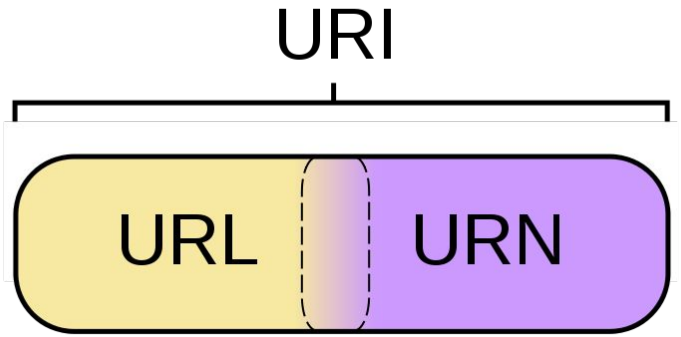
Resources

Represent

Representations

```
<html>
<head>
<title>
Home:: Michael L. Nelson, Old Dominion University
</title>
<link rel="stylesheet" type="text/css" href="mln.css"/>
<script type="text/javascript" src="mln.js"></script>
…
```

What's a URx?



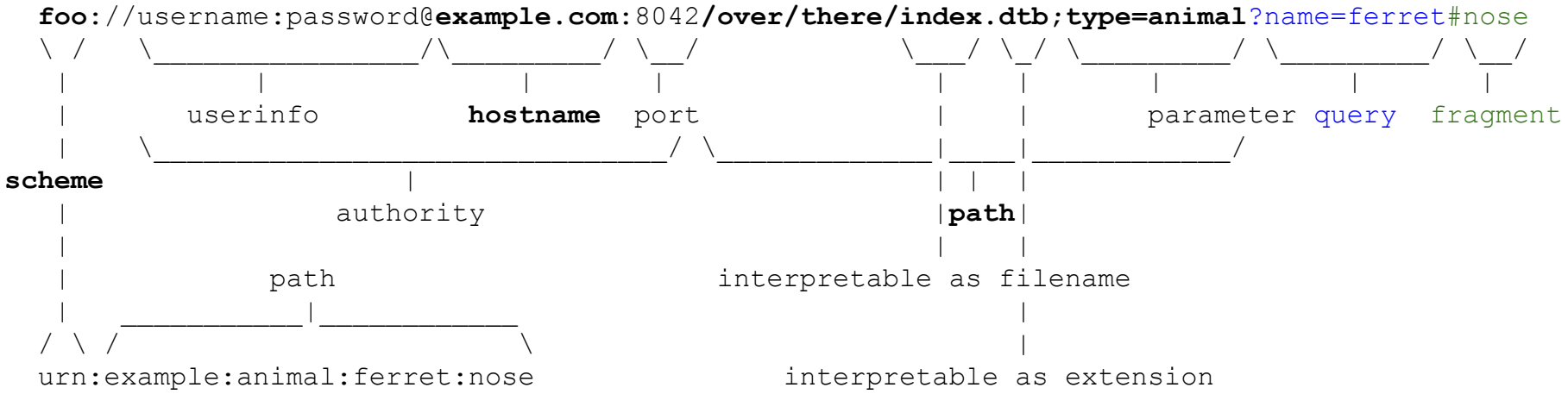
URI (identifier) - String of characters used to identify a name or resource on the Internet

URL (locator) - Where to find a resource

URN (name) - Name of a resource

Figure source: URI Euler Diagram (Wikipedia)

URI Components



Other examples:

- http://example.org/absolute/path/to/resource.txt
- ftp://example.org/resource.txt
- urn:issn:1535-3613

Figure source: Uniform Resource Identifier (Wikipedia)

Web Science:
Web Science and Web Architecture
(Part 4 - Talking to Web Servers)

CS 432/532

Old Dominion University

Permission has been granted to use these slides from Frank McCown, Michael L. Nelson, Alexander Nwala, Michele C. Weigle

Talking to HTTP servers…

```
% curl --head https://www.cs.odu.edu/~mweigle/
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 03 Jan 2020 18:12:10 GMT
Content-Type: text/html; charset=ISO-8859-1;
Connection: keep-alive
Expires: Tue, 01 Jan 2002 00:00:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Front-End-Https: on
```

How To Use - curl

```
% curl -I http://www.google.com/
HTTP/1.1 200 OK
Cache-Control: private, max-age=0
Date: Mon, 12 Jan 2009 15:45:57 GMT
Expires: -1
Content-Type: text/html; charset=ISO-8859-1
Set-Cookie: PREF=ID=9a80d3f602b685f3:TM=1231775157:LM=1231775157:S=imGxRyNsTD0Zczm5;
expires=Wed, 12-Jan-2011 15:45:57 GMT; path=/; domain=.google.com
Server: gws
Content-Length: 0
```

more curl…

```
% curl https://www.cs.odu.edu/~mweigle/
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!--<html xmlns="http://www.w3.org/1999/xhtml" $GmaIEFix>-->
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Home | Michele C. Weigle </title>
  <meta http-equiv='Conte
...
```

```
% curl -i https://www.cs.odu.edu/~mweigle/
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 03 Jan 2020 18:11:04 GMT
Content-Type: text/html; charset=ISO-8859-1;
Transfer-Encoding: chunked
Connection: keep-alive
Expires: Tue, 01 Jan 2002 00:00:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Vary: Accept-Encoding
Front-End-Https: on

<!DOCTYPE html
...
```

wget

```
% wget https://www.cs.odu.edu/~mweigle/
--2020-05-25 14:48:26--  https://www.cs.odu.edu/~mweigle/
Resolving www.cs.odu.edu (www.cs.odu.edu)... 128.82.4.161
Connecting to www.cs.odu.edu (www.cs.odu.edu)|128.82.4.161|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html'

index.html                [ <=>                 ]  20.92K  --.-KB/s in 0.005s

2020-05-25 14:48:27 (3.95 MB/s) - 'index.html' saved [21420]
```

See: curl vs Wget

curl and wget are useful, but issuing raw HTTP requests is more fun...

GET

```
% telnet example.com 80
Trying 93.184.216.34...
Connected to example.com.
Escape character is '^]'.
GET / HTTP/1.1
Host: example.com
Connection: close
```

Request
(ends w/ CRLF)

```
HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Fri, 03 Jan 2020 18:33:26 GMT
Etag: "3147526947+gzip"
Expires: Fri, 10 Jan 2020 18:33:26 GMT
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
Server: ECS (dcb/7EEB)
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 1256
Connection: close

<!doctype html>
<html>
...
Connection closed by foreign host.
```

Port 80 is the default HTTP port

Response
(CRLF separates header from body)

This doesn't work for HTTPS

Port 443 is the default HTTPS port

```
% telnet www.cs.odu.edu 443
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
GET / HTTP/1.1
Host: www.cs.odu.edu
Connection: close

HTTP/1.1 400 Bad Request
Server: nginx
Date: Fri, 03 Jan 2020 18:54:33 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: close

108
<html>
<head><title>400 The plain HTTP request was sent to HTTPS port</title></head>
<body bgcolor="white">
<center><h1>400 Bad Request</h1></center>
<center>The plain HTTP request was sent to HTTPS port</center>
<hr><center>nginx</center>
</body>
</html>

0

Connection closed by foreign host.
```

In HTTPS, HTTP is tunnelled inside encrypted layer, so no plain-text

HEAD

```
% telnet example.com 80
Trying 93.184.216.34...
Connected to example.com.
Escape character is '^]'.
HEAD / HTTP/1.1
Host: example.com
Connection: close

HTTP/1.1 200 OK
Content-Encoding: gzip
Accept-Ranges: bytes
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Fri, 03 Jan 2020 18:39:17 GMT
Etag: "3147526947+gzip"
Expires: Fri, 10 Jan 2020 18:39:17 GMT
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
Server: ECS (dcb/7EEF)
X-Cache: HIT
Content-Length: 648
Connection: close

Connection closed by foreign host.
```

OPTIONS

```
% telnet example.com 80
Trying 93.184.216.34...
Connected to example.com.
Escape character is '^]'.
OPTIONS / HTTP/1.1
Connection: close
Host: example.com

HTTP/1.1 200 OK
Allow: OPTIONS, GET, HEAD, POST
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Fri, 03 Jan 2020 18:41:43 GMT
Expires: Fri, 10 Jan 2020 18:41:43 GMT
Server: EOS (vny006/044E)
Content-Length: 0
Connection: close

Connection closed by foreign host.
```

```
% telnet awoiaf.westeros.org 80
Trying 108.162.197.188...
Connected to awoiaf.westeros.org.
Escape character is '^]'.
POST /index.php/Special:Export HTTP/1.1
Host: awoiaf.westeros.org
Content-type: text/plain
Content-length: 10

123456789
```

```
HTTP/1.1 200 OK
Server: cloudflare-nginx
Date: Wed, 28 Aug 2013 15:25:53 GMT
Content-Type: text/html; charset=utf-8
Content-language: en
X-Frame-Options: DENY
Vary: Accept-Encoding, Cookie
[lots of headers deleted]

<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
[lot of html deleted]
```

POST

Request
(CRLF separates
header from body)

Response
(CRLF separates
header from body)

Finding POST in HTML…

```
…
<p>In the latter case you can also use a link, for example
<a href="/index.php/Special:Export/Main_Page" title="Special:Export/Main
Page">Special:Export/Main Page</a> for the page"
<a href="/index.php/Main_Page" title="Main Page">Main Page</a>". </p>
<form method="post" action="/index.php?title=Special:Export&amp;action=submit">
<label for="catname">Add pages from category:</label> 
<input name="catname" size="40" id="catname" class="mw-ui-input"/> 
<input name="addcat" type="submit" value="Add"/><br/>
<textarea name="pages" cols="40" rows="10"></textarea><br/>
<input name="curonly" type="checkbox" value="1" checked="checked" id="curonly"/> 
<label for="curonly">Include only the current revision, not the full history</label>
<br/><input name="templates" type="checkbox" value="1" id="wpExportTemplates"/> 
<label for="wpExportTemplates">Include templates</label><br/>
<input name="wpDownload" type="checkbox" value="1" checked="checked" id="wpDownload"/> 
<label for="wpDownload">Save as file</label><br/>
<input title="[s]" accesskey="s" type="submit" value="Export"/></form>
…
```

HTTP Response Codes

not "error" codes!

- 1xx: Informational - Request received, continuing process
- 2xx: Success - The action was successfully received, understood, and accepted
- 3xx: Redirection - Further action must be taken in order to complete the request
- 4xx: Client Error - The request contains bad syntax or cannot be fulfilled
- 5xx: Server Error - The server failed to fulfill an apparently valid request

from Section 6 of RFC 7231

501 - Not Implemented

```
% telnet awoiaf.westeros.org 80
Trying 104.26.6.227...
Connected to awoiaf.westeros.org.
Escape character is '^]'.
NOTAREALMETHOD /index.html HTTP/1.1
Connection: close
Host: awoiaf.westeros.org

HTTP/1.1 501 Not Implemented
Date: Fri, 03 Jan 2020 18:48:14 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: close
Set-Cookie: __cfduid=de0777e48aa880e4c73812856018a23121578077294; expires=Sun, 02-Feb-20 18:48:14 GMT; path=/; domain=.westeros.org;
HttpOnly; SameSite=Lax
CF-Cache-Status: DYNAMIC
Server: cloudflare
CF-RAY: 54f7254ec90256af-IAD

15d
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
         "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>501 - Not Implemented</title>
</head>
<body>
<h1>501 - Not Implemented</h1>
</body>
</html>

0

Connection closed by foreign host.
```

301 - Moved Permanently

```
% telnet awoiaf.westeros.org 80
Trying 104.26.7.227...
Connected to awoiaf.westeros.org.
Escape character is '^]'.
OPTIONS / HTTP/1.1
Connection: close
Host: awoiaf.westeros.org

HTTP/1.1 301 Moved Permanently
Date: Fri, 03 Jan 2020 19:05:05 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: close
Set-Cookie: __cfduid=ddd3b69f46dd1c9ad7774cbeb1ab3e14f1578078305; expires=Sun, 02-Feb-20
19:05:05 GMT; path=/; domain=.westeros.org; HttpOnly; SameSite=Lax
X-Content-Type-Options: nosniff
Set-Cookie: PHPSESSID=cas9kgnt1miiple9u2dhand5i3; path=/
Pragma: no-cache
Vary: Accept-Encoding, Cookie
Last-Modified: Fri, 03 Jan 2020 19:05:05 GMT
Location: https://awoiaf.westeros.org/index.php/Main_Page
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Cache-Control: private, max-age=0, s-maxage=300
Accept-Ranges: bytes
X-Cache: MISS
...
```

Multiple redirects possible!

```
% curl -I -L https://t.co/Nbleumtera
HTTP/1.1 301 Moved Permanently
cache-control: private,max-age=300
content-length: 0
date: Fri, 03 Jan 2020 19:09:53 GMT
expires: Fri, 03 Jan 2020 19:14:53 GMT
location: http://bit.ly/2QIY1jW
server: tsa_a
set-cookie: muc=d8cec1cd-a7d1-40db-8d76-737b85d13f08; Max-Age=63072000; Expires=Sun, 2 Jan 2022 19:09:53 GMT; Domain=t.co
strict-transport-security: max-age=0
vary: Origin
x-connection-hash: fc0224ab9bcbfa6261c63e01389c5472
x-response-time: 8

HTTP/1.1 301 Moved Permanently
Server: nginx
Date: Fri, 03 Jan 2020 19:09:53 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 153
Cache-Control: private, max-age=90
Location: https://www.odu.edu/news/2019/10/former_odu_pitcher_i#.Xg3_3RdKigR
Via: 1.1 google

HTTP/1.1 200 OK
Date: Fri, 03 Jan 2020 19:09:53 GMT
Server: Apache/2.4.6 (Red Hat Enterprise Linux)
Vary: Host
Accept-Ranges: bytes
Connection: close
Content-Type: text/html; charset=UTF-8
Set-Cookie: BIGipServerWEB_HTTPS_PROD.app~WEB_HTTPS_PROD_pool_campus=rd627o00000000000000000ffff8052619fo80; path=/;
```

302 - Found

```
% curl -I https://dx.doi.org/10.1145/1998076.1998100
HTTP/2 302
date: Wed, 09 Sep 2020 21:54:18 GMT
content-type: text/html;charset=utf-8
content-length: 195
set-cookie: __cfduid=df8601cbe1c37abae1beb49becc95d4ac1599688458; expires=Fri,
09-Oct-20 21:54:18 GMT; path=/; domain=.doi.org; HttpOnly; SameSite=Lax; Secure
vary: Accept
location: http://portal.acm.org/citation.cfm?doid=1998076.1998100
expires: Wed, 09 Sep 2020 22:32:29 GMT
cf-cache-status: DYNAMIC
cf-request-id: 051675da7a0000255023bec200000001
expect-ct: max-age=604800,
report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"
strict-transport-security: max-age=31536000; includeSubDomains; preload
server: cloudflare
cf-ray: 5d0425a3f83e2550-IAD
```

303 - See Other

```
% telnet dx.doi.org 80
Trying 38.100.138.149...
Connected to dx.doi.org.
Escape character is '^]'.
HEAD http://dx.doi.org/10.1007/978-3-642-24469-8_16 HTTP/1.1
Host: dx.doi.org
Connection: close

HTTP/1.1 303 See Other
Server: Apache-Coyote/1.1
Location:  http://www.springerlink.com/index/10.1007/978-3-642-24469-8_16
Expires: Wed, 11 Jan 2012 12:04:29 GMT
Content-Type: text/html;charset=utf-8
Content-Length: 210
Date: Tue, 10 Jan 2012 17:56:41 GMT
Connection: close
```

Example from 2012, before move to HTTPS

404 - Not Found

```
% telnet www.cs.odu.edu 80
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
HEAD /lasdkfjalsdkfjldaskfj HTTP/1.1
Host: www.cs.odu.edu
Connection: close

HTTP/1.1 404 Not Found
Server: nginx
Date: Fri, 03 Jan 2020 19:13:06 GMT
Content-Type: text/html
Connection: close

Connection closed by foreign host.
```

```
% curl -I https://www.cs.odu.edu/lasdkfjalsdkfjldaskfj
HTTP/1.1 404 Not Found
Server: nginx
Date: Fri, 11 Sep 2020 18:46:37 GMT
Content-Type: text/html; charset=iso-8859-1
Connection: keep-alive
```

400 - Bad Request

```
% telnet www.cs.odu.edu 80
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
HEAD http://www.cs.odu.edu/~mln/ HTTP/1.1
Connection: close

HTTP/1.1 400 Bad Request
Server: nginx
Date: Fri, 03 Jan 2020 19:18:39 GMT
Content-Type: text/html
Connection: close

Connection closed by foreign host.
```

Missing required header

```
Host: www.cs.odu.edu
```

Objectives

- List the main interdisciplinary components of web science
- Describe the small world network phenomenon and how it relates to web science
- Given a set of pages and their links, classify each page as part of the SCC, IN, OUT, Tendrils, Tubes, or Disconnected categories of the Bow-Tie Structure of the Web.
- Explain the difficulties in determining the size of the Web.
- Describe the operation of a web crawler.
- Describe the steps required to load a typical web page, in terms of application-layer (DNS, HTTP) networking operations required.
- Differentiate between a web resource and web representation.
- Demonstrate how to communicate with a web server using curl, wget, and telnet.
- Describe the different categories of HTTP response codes
- Explain how a web client knows what URI to request next upon receiving a response with a 3xx (redirection) status code.