

# The Internet, the Web, and HTML

Jamie Saxon

Introduction to Programming for Public Policy

Halloween 2016



# **The Internet and the World Wide Web**

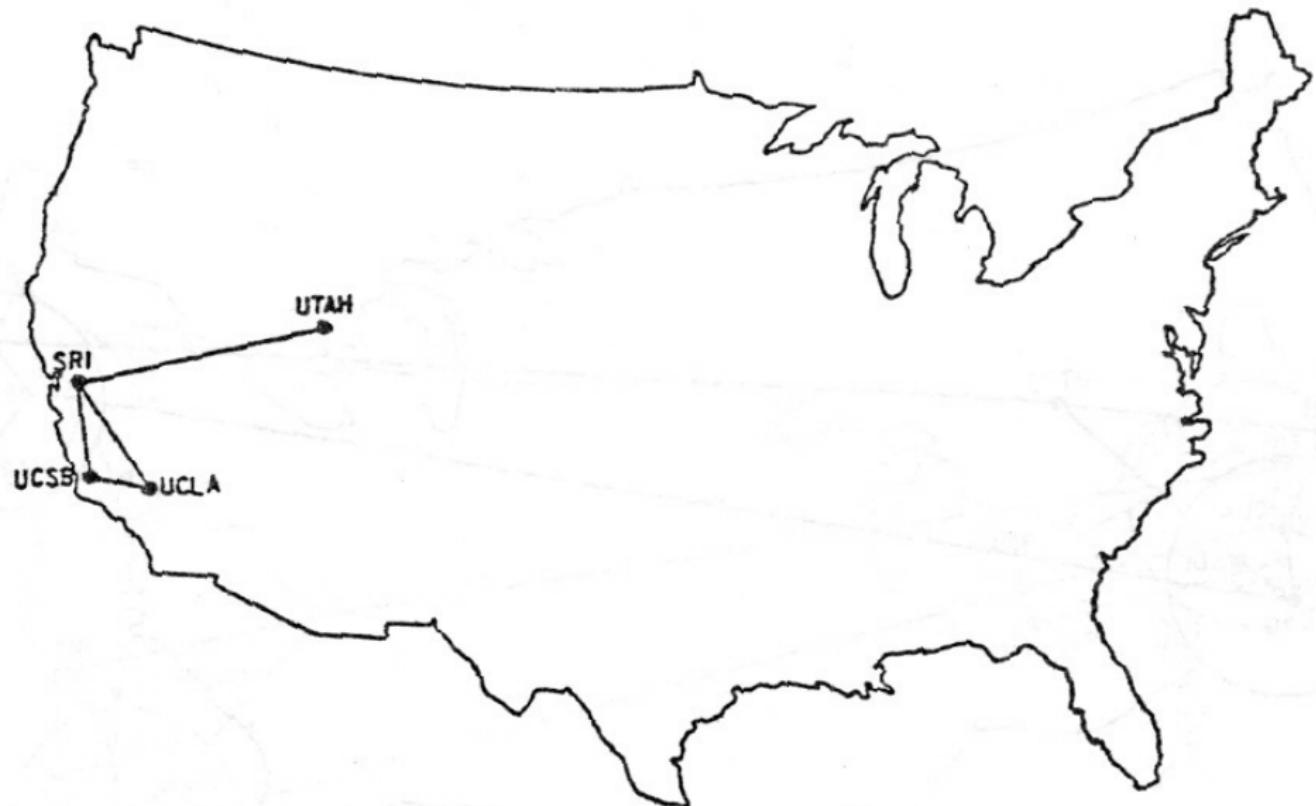
# Fundamental Definitions

The **Internet** is a physical network, and a set of protocols for shipping information across that network.

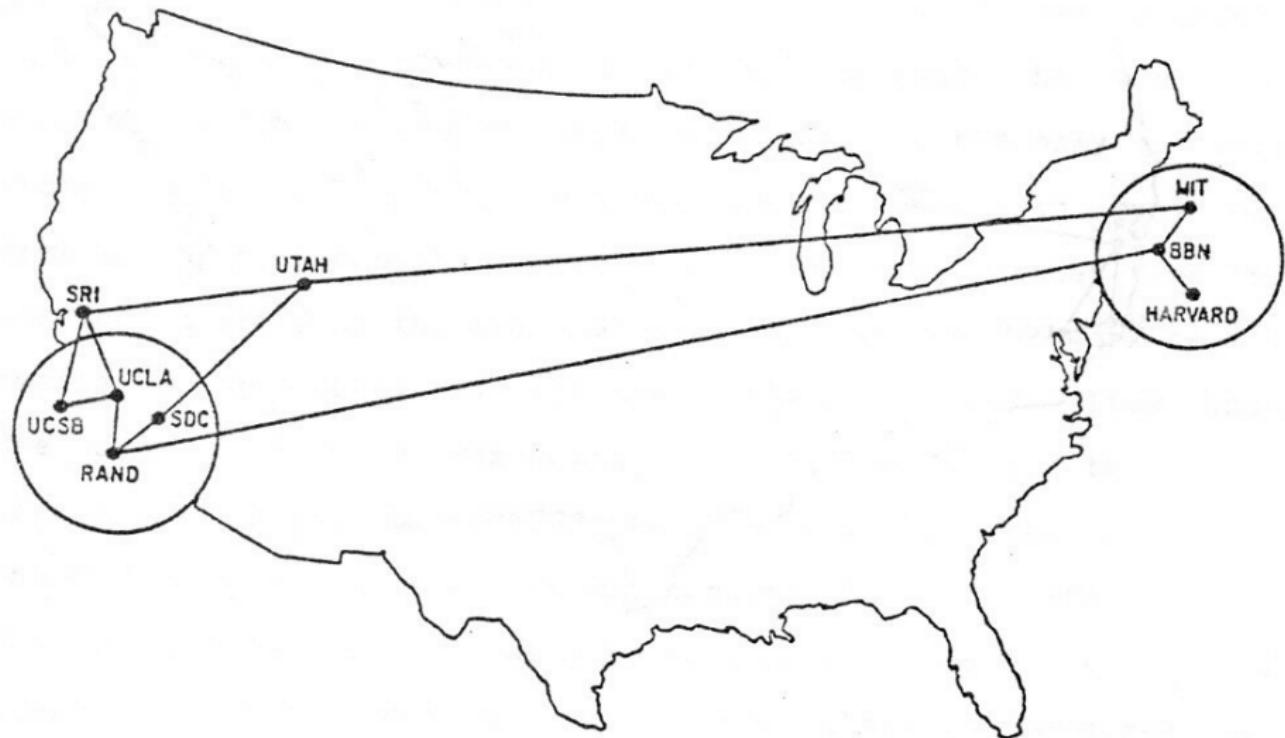
The **World Wide Web** uses a high-level protocol, called the Hyper Text Transfer Protocol (http), along with a markup language (html), to transfer interlinked documents across the Internet.

# History of the Internet in One Slide

- ▶ Started as a Defense Advanced Research Projects Agency (DARPA/DoD) ARPANET project to interconnect computers.
- ▶ First transmission between nodes at UCLA in 1969; by 1970 it reached across the US (to Boston).
- ▶ Transmission Control Protocol/Internet Protocol (TCP/IP, slide 13) developed throughout the 1970s.
  - ▶ Became sole internet protocol in 1983.
- ▶ ARPANET declared 'operational' (no-longer research) in 1975; transferred to the military.
- ▶ Ethernet standard (transmission on wires) written in 1981.
- ▶ In the 1980s, the internet shifted to the National Science Foundation; in 1983, it switched to its modern protocol (TCP/IP).
  - ▶ NSFNET provided the backbone of the internet from 1985 to 1995.
- ▶ Clinton I privatized the internet backbone in 1994.

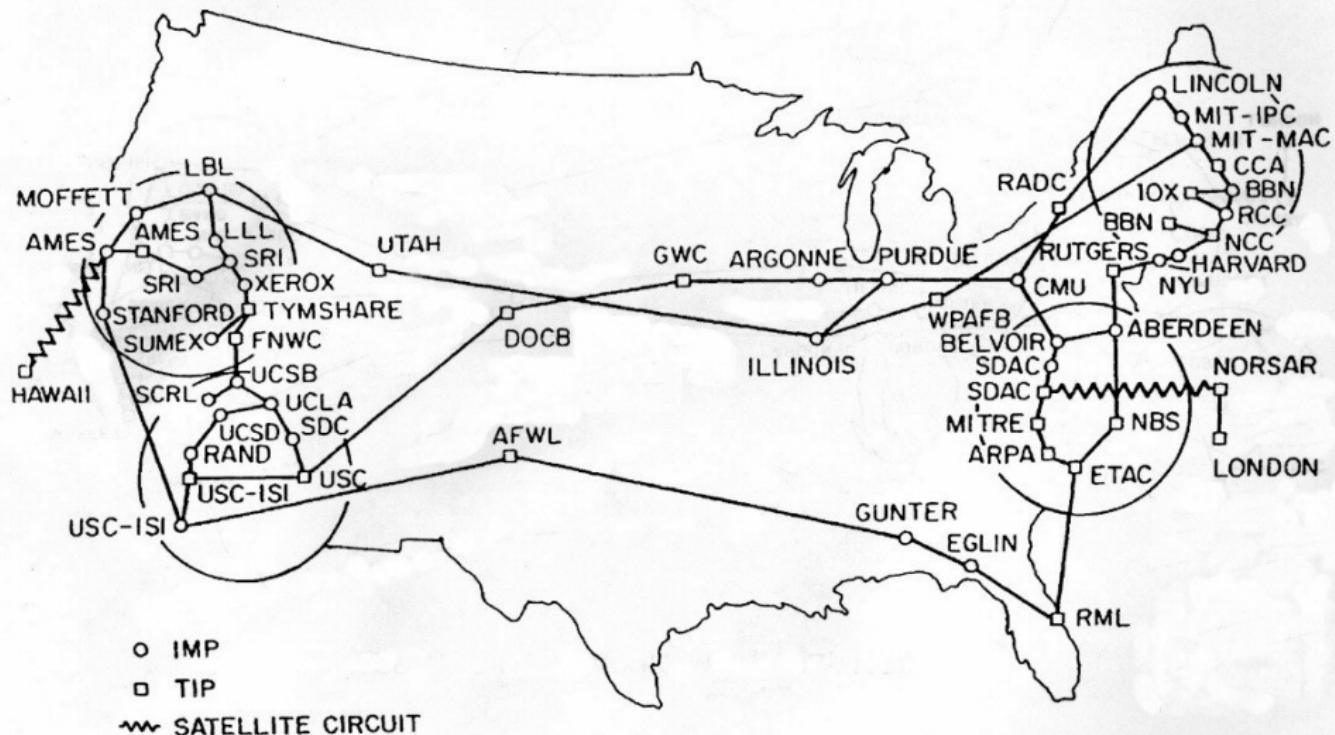


The ARPANET in December 1969

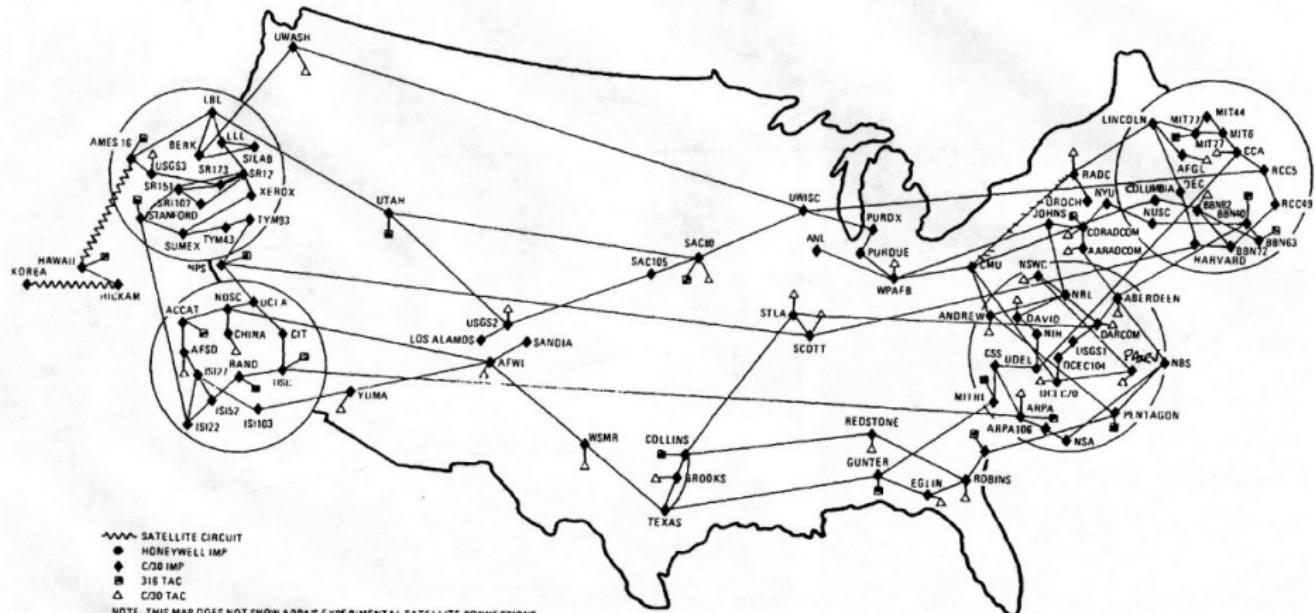


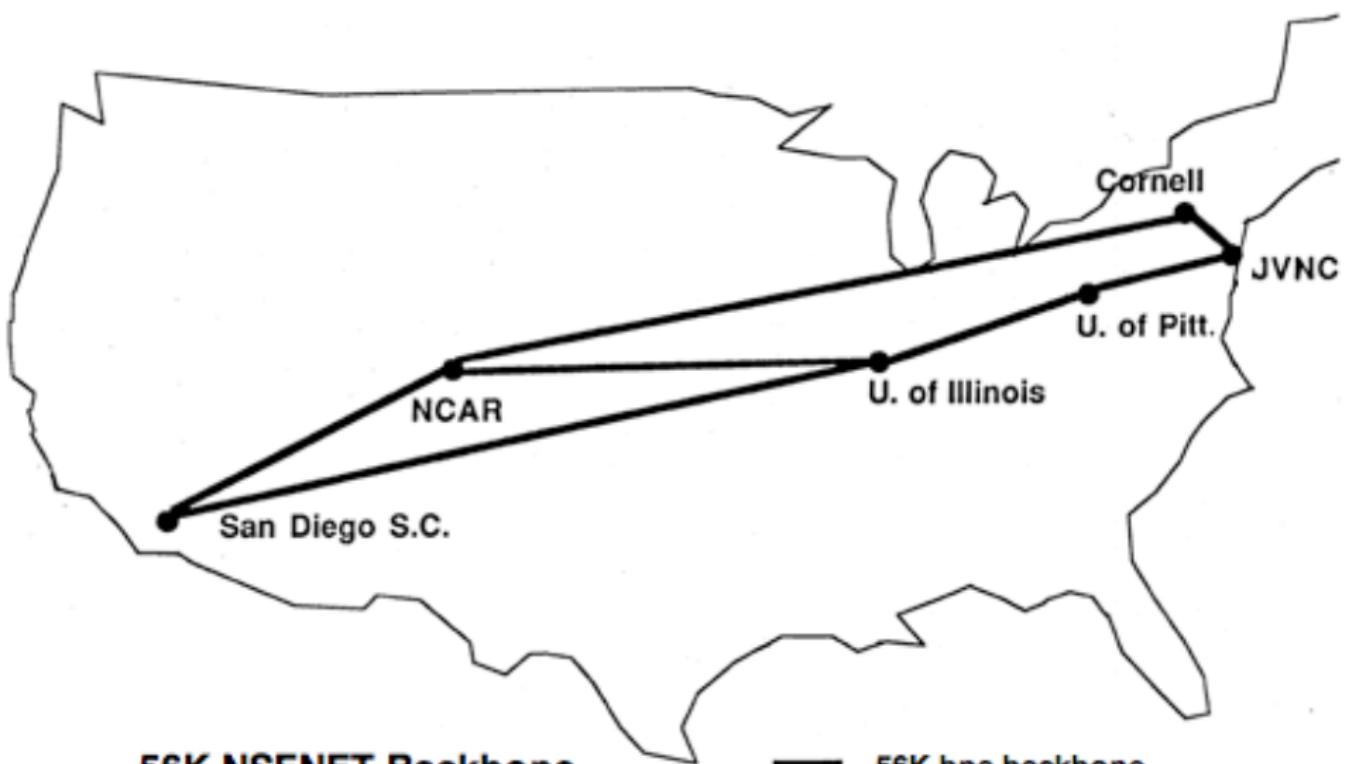
# ARPA NETWORK, GEOGRAPHIC MAP

JUNE 1975



## ARPANET GEOGRAPHIC MAP, SEPTEMBER 1983

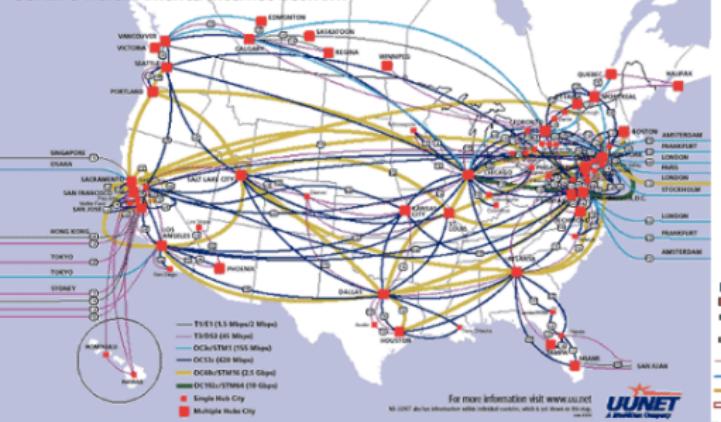




**56K NSFNET Backbone**  
July 1986 - July 1988

— 56K bps backbone  
● Fuzzball nodes

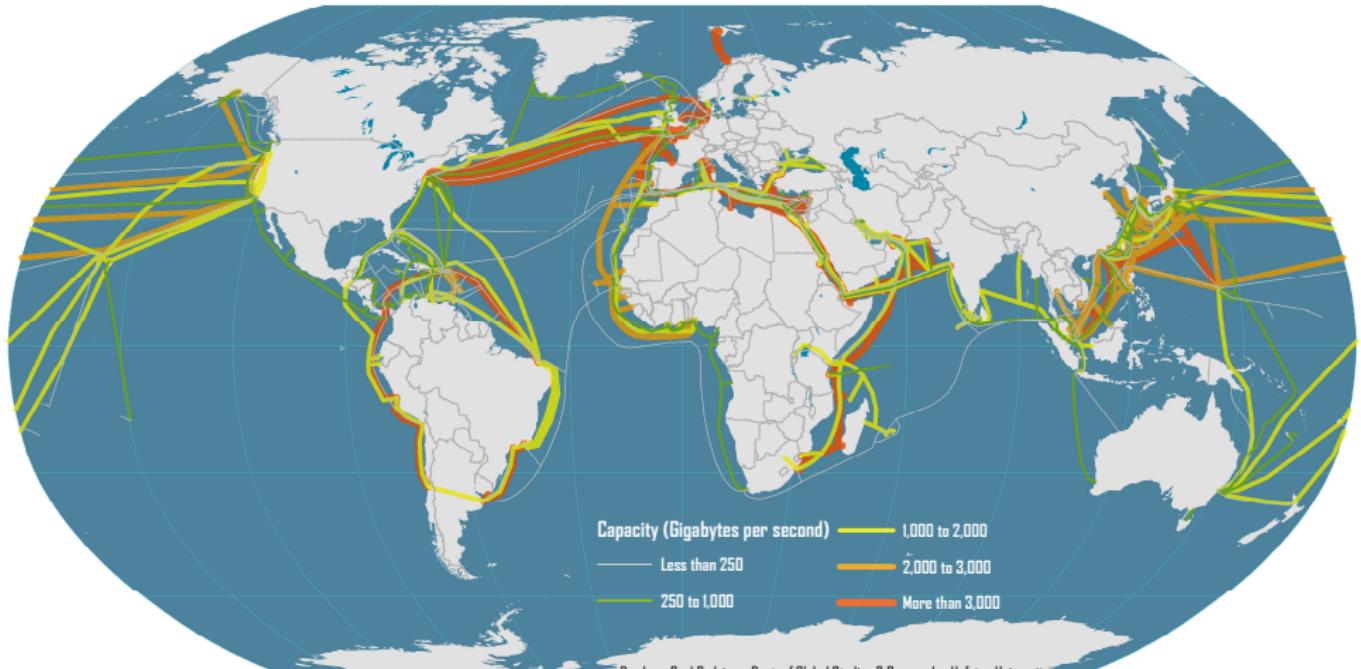
## UUNET's North America Internet network



## AT&T IP BACKBONE NETWORK



## Global Submarine Cable Network



Dr. Jean-Paul Rodrigue, Dept. of Global Studies & Geography, Hofstra University.

Source: Dataset encoded by Greg Mahlknecht, <http://www.cablemap.info>



# The Internet Protocol Suite: Moving Packets

How do the packets actually get divided up and sent around?

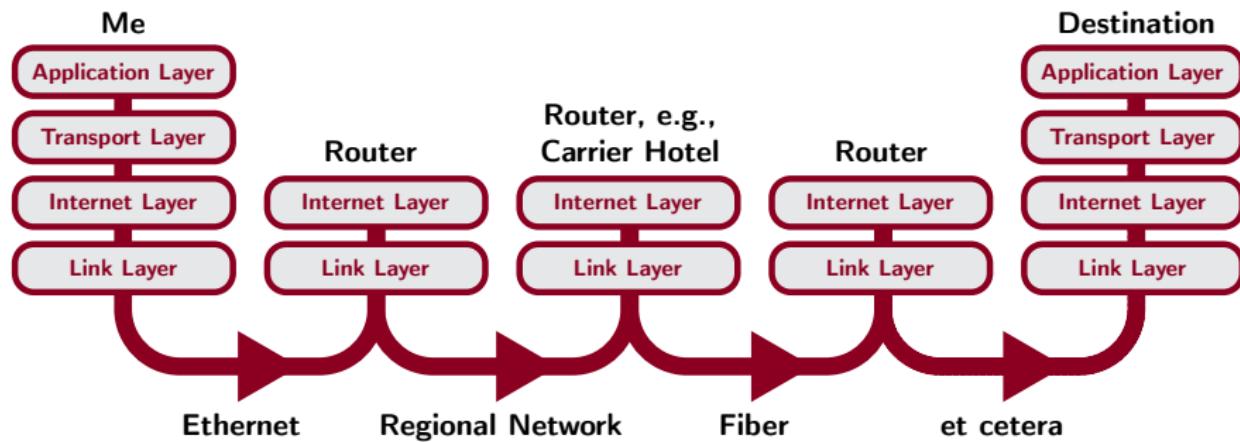
- ▶ Higher-level applications deliver payloads to lower-level protocols, that ensure deliver, address, the packets, and move them across wires.
  - ▶ The high-level protocol just needs to know the IP address it wants, and the port at the destination.
- ▶ Together, these make up the Internet Protocol Suite, or TCP/IP.
  - ▶ Often conceptualized as four layers.\*

<b>Application Layer</b>	High-level internet apps.	HTTP, SMTP, POP, DNS, etc.
<b>Transport Layer</b>	Wrapping packets; ensuring delivery.	Transmission Control Protocol, UDP.
<b>Internet Layer</b>	Directions, addressing, mapping.	Internet Protocol, ICMP, IGMP.
<b>Link Layer</b>	Physical medium (ethernet).	Ethernet, Wi-Fi, ARP.

\*A competing model, the OSI model, uses 7.

# Moving Across the Internet

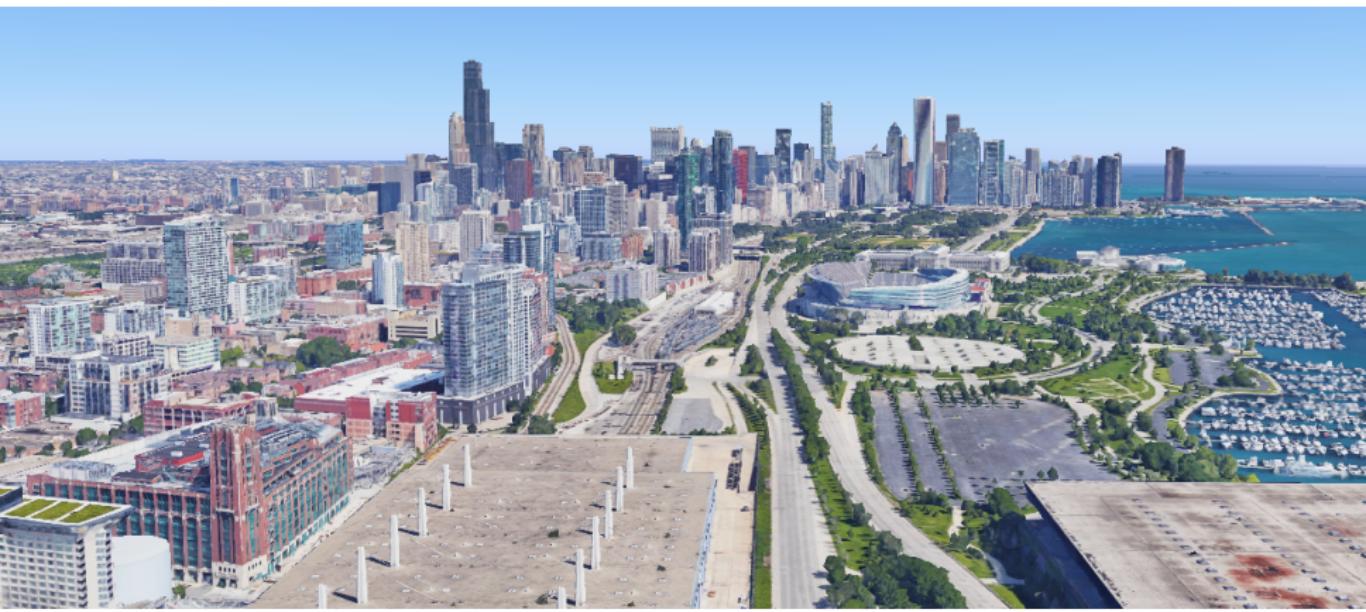
Data is divided, wrapped and delivered to lower levels, where they are repeatedly routed (internet layer) and shipped (over physical cables).

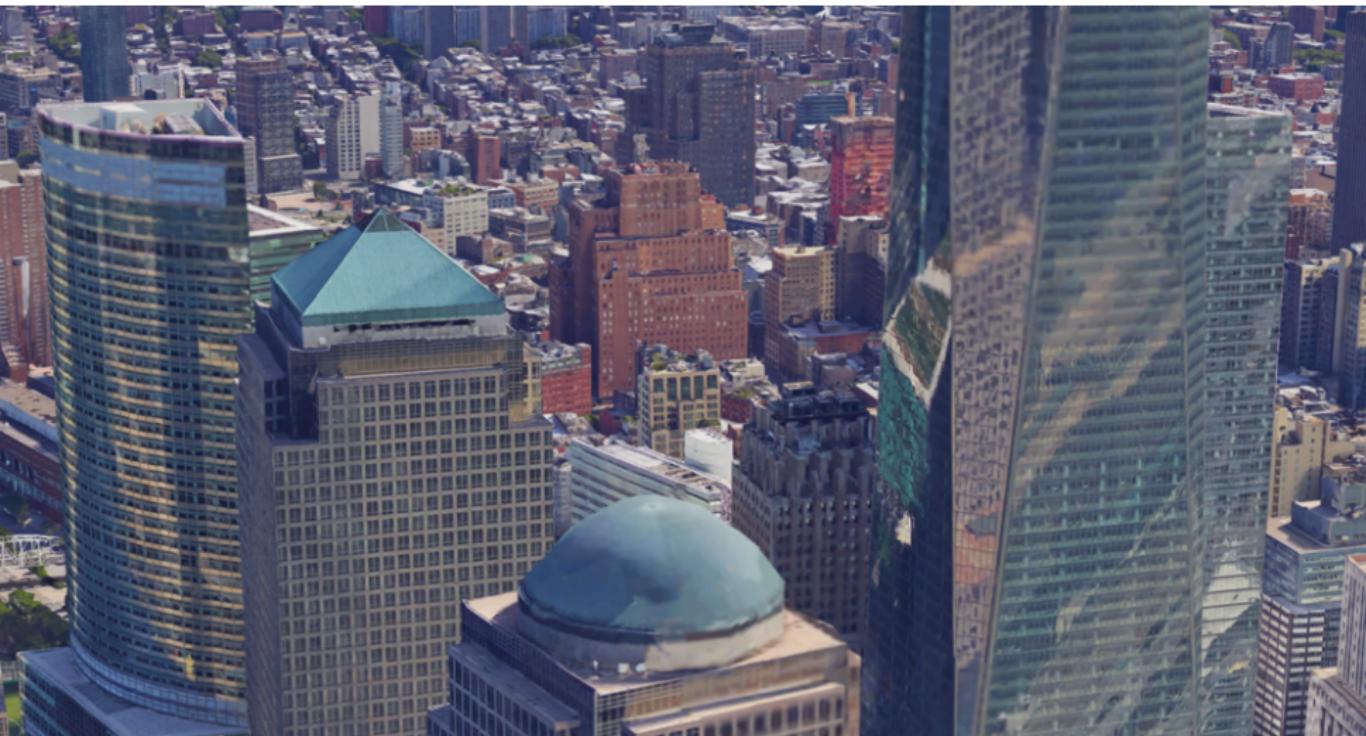


# Mapping from Here to Geneva

Hop	IP	Where	Time
1	10.0.0.1	(My Router)	-
2	96.120.26.81	Comcast: Mount Prospect, IL	11.266 ms
3	elmhurst.il.chicago.comcast.net	Comcast: Elmhurst, IL	9.386 ms
4	162.151.92.89	Comcast: Lower West Side	12.537 ms
5	area4.il.chicago.comcast.net	Comcast: Lower West Side	11.665 ms
6	350ecermak.il.ibone.comcast.net	350 East Cermak, Chicago	13.547 ms
7	newyork.ny.ibone.comcast.net	60 Hudson St, NYC	32.365 ms
8	111eighthave.ny.ibone.comcast.net	Google Building, NYC	30.809 ms
9	111eighthave.ny.ibone.comcast.net	Google Building, NYC	33.153 ms
10	217.239.43.33	Deutsche Telekom, Remscheid	121.475 ms
11	193.159.166.222	Deutsche Telekom, Monchengladbach	238.268 ms
12	e513-e-rbrxl-2-ne0.cern.ch	CERN, Geneva	119.991 ms

traceroute to CERN/Geneva



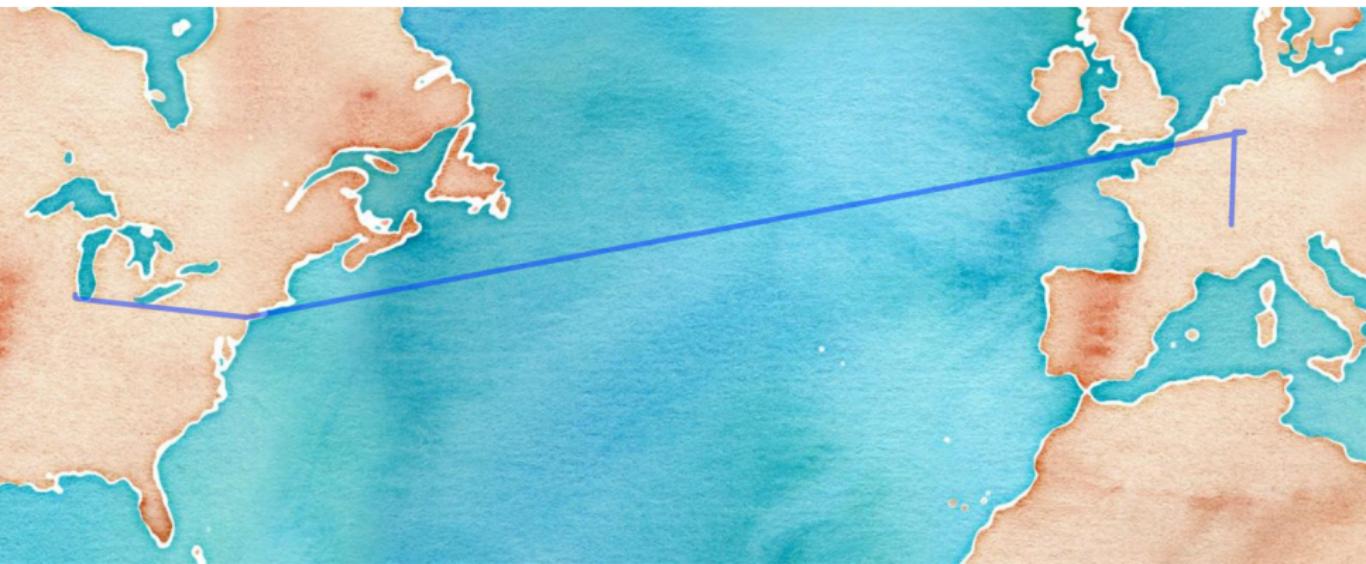




Google

THE PORT OF NEW YORK AUTHORITY INLAND TERMINAL NUMBER ONE





# Hyper Text Transfer Protocol



# The Hyper Text Transfer Protocol

- ▶ Most of the data that you access on the internet comes over http or https (secure).
- ▶ A web browser is a ‘client,’ issuing requests through a TCP/IP port.
- ▶ A ‘server’ is an application on a computer, listening on a port.
- ▶ HTTP provides a framework for these requests and responses.
  - ▶ HTTP takes the transmission across the Internet at the lower levels (TCP/IP/Link) for granted.

# GET and POST Requests

- ▶ The two most-common request types are GET and POST.
- ▶ Both request resources with a 'uniform resource locator' (URL):

```
scheme://host[:port]/path[?query][#fragment]
```

- ▶ GET simply requests a URL.
- ▶ POST additionally sends data to the server in the request 'body.'
- ▶ There are also several methods used more-commonly in e.g., javascript (not covered): HEAD, PUT, DELETE, etc.

# The Response/Resource

- ▶ The server responds to the request with a status code and, hopefully, the requested resource.
- ▶ The resource returned may be...
  - ▶ static: a file &c. that is already prepared.
  - ▶ dynamic: using limited data from the name of the requested resource (GET) or data from the body of the request (POST) the server may create or modify a resource 'on-the-fly' (next week).
- ▶ The most common status code is **200**/OK, but you've probably also seen **404**/Not Found, and experienced **301**/Moved or **307**/redirect.
- ▶ Most common type of document is an **html file**, which may include references (URLs) for additional resources (pictures, style, etc.) that the browser must also request, and which the server will supply.

# Hyper Text Markup Language

We're covering this so that:

1. You can make simple websites to share data.
2. You can sift through other people's websites to grab their data.

# Webpages

- ▶ Modern webpages have myriad components – pictures, videos, text, fonts and styling (css), links, databases, scripts, authentication, etc.
- ▶ These components are drawn together by documents in Hyper Text Markup Language (html).
  - ▶ HTML ‘tags’ define headers, lists, images, links, tables, etc.
  - ▶ The browser parses these tags (and css and javascript code).
- ▶ Presentation of elements specified by ‘Cascading Style Sheets.’
- ▶ The totality of interlinking html documents is the web.

# The Fundamental HTML Tags<sup>†</sup>

- ▶ Most of these come in pairs: <tag>value</tag>.
- ▶ Table, break, and img are the exceptions.

Tag	Meaning
<title>	Page Title
<header>	Header Content
<body>	Body of the page
<h1>	Header (Highest)
<h2-6>	Header (Lower)
<p>,  	Paragraph, Break
<em>, <b>	Emphasis/Bold
<ol>, <ul>, <li>	(Un)order List, List Element
<a href="">	Hyperlink
<table>, <tr>, <th>, <td>	Table, row, header, and cell
<img src="">	Image (single tag)
<div>	div: ~ block

<sup>†</sup>We will, perhaps, make it to forms next week.

# An HTML Skeleton

```
<!DOCTYPE html>
<html>
<head>
    <title>My Title</title>
</head>
```

```
<body>
```

```
<h1>My Site</h1>
```

Some information about my site.

```
</body>
</html>
```

# HTML Example

- ▶ Go find this skeleton at:

<https://harris-ippp.github.io/lectures/skeleton.html>

or

<http://tinyurl.com/gmqnwzz>

- ▶ Save page as; Format 'Web Page, HTML only'.
- ▶ Navigate to it using the standard directory system, and open it in both Atom and your browser.
- ▶ We'll use this to write an 'informative' article on chickens, [here](#).

## You'll Never Guess What *Gallus gallus domesticus* Can Do For Your Health

I am so happy to eat chicken, it just tastes so *gosh-darned good!*

There are many proven health benefits to eating chicken including:

Benefit	Verified
Calories	True
Endorphins	True
Sleeping late	True
Ice Cream	(Under investigation)
Energy Neutral	False



At right are some chicks, striving for tenderness and juiciness. There is also a chicken lady and her gentleman [here](#).

**Consectetur adipiscing elit**, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# CSS

Once your content is in place (or you've given up on content), format it!

- ▶ CSS rules apply to all objects of one (or many) types:

```
h1, h2, h3 { color: darkred; font-weight: bold; }
```

- ▶ Rules apply hierarchically: for instance, a link in an item in a list would be “ul li a.”
- ▶ Classes of element types are referred to with a dot: ul.nav.
- ▶ Elements may be accessed by ‘id’ with a hash: #myelement.
- ▶ Specify :visited links, or elements the mouse is :hover-ing over.

# A Few Common Names

- ▶ There are too many property names to remember them all, but here are some of the most common ones.
  - ▶ Find others and check meanings [here](#) or by googling.
- ▶ Will discuss how spacing works, next.

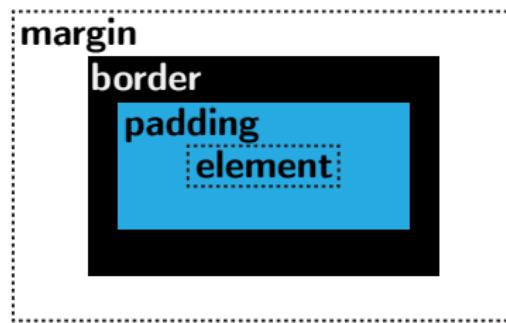
---

background	border-spacing	display	margin
background-color	border-style	float	padding
border	border-width	font-family	text-align
border-color	color	font-weight	text-decoration

---

# CSS Spacing

- ▶ There are three attributes for spacing: padding, border, and margin.
- ▶ These maybe specified as pixels (px) 'em' (m-width), 'ex' (x-width), or as a percent of the parent element (%).
- ▶ A single value corresponds to all borders, two values correspond to top/bottom and side borders, and four is (top, right, bottom, left)



# Writing CSS or Including a File

- ▶ CSS rules may be added in the header, between `<script>` tags.
- ▶ A file of .css rules in your directory may be loaded via  
`<link rel="stylesheet" href="my_style.css">`

# A CSS ‘Trick’ for Navigation from a List

Common to build navigation from a list of links:

- ▶ Display list elements on a single line (`inline`).
- ▶ Display links as `inline-block`: respect their padding and margins.

```
ul.nav li { display: inline; }  
ul.nav li a { display: inline-block; }
```