



College of Information Studies

University of Maryland Hornbake Library Building College Park, MD 20742-4345

---

# Practicum: Networks, Basic HTML

Week 2

LBSC 690

Information Technology

# Muddiest Points

- Bytes, nanos, gigas, Hz, ...
- The memory hierarchy
- Why disks scatter things around
- RAID arrays
- The term project

# Goals for Today

- Understand how bits get from here to there
- Learn to move files to a “Web server”
- Use those skills to make a Web page

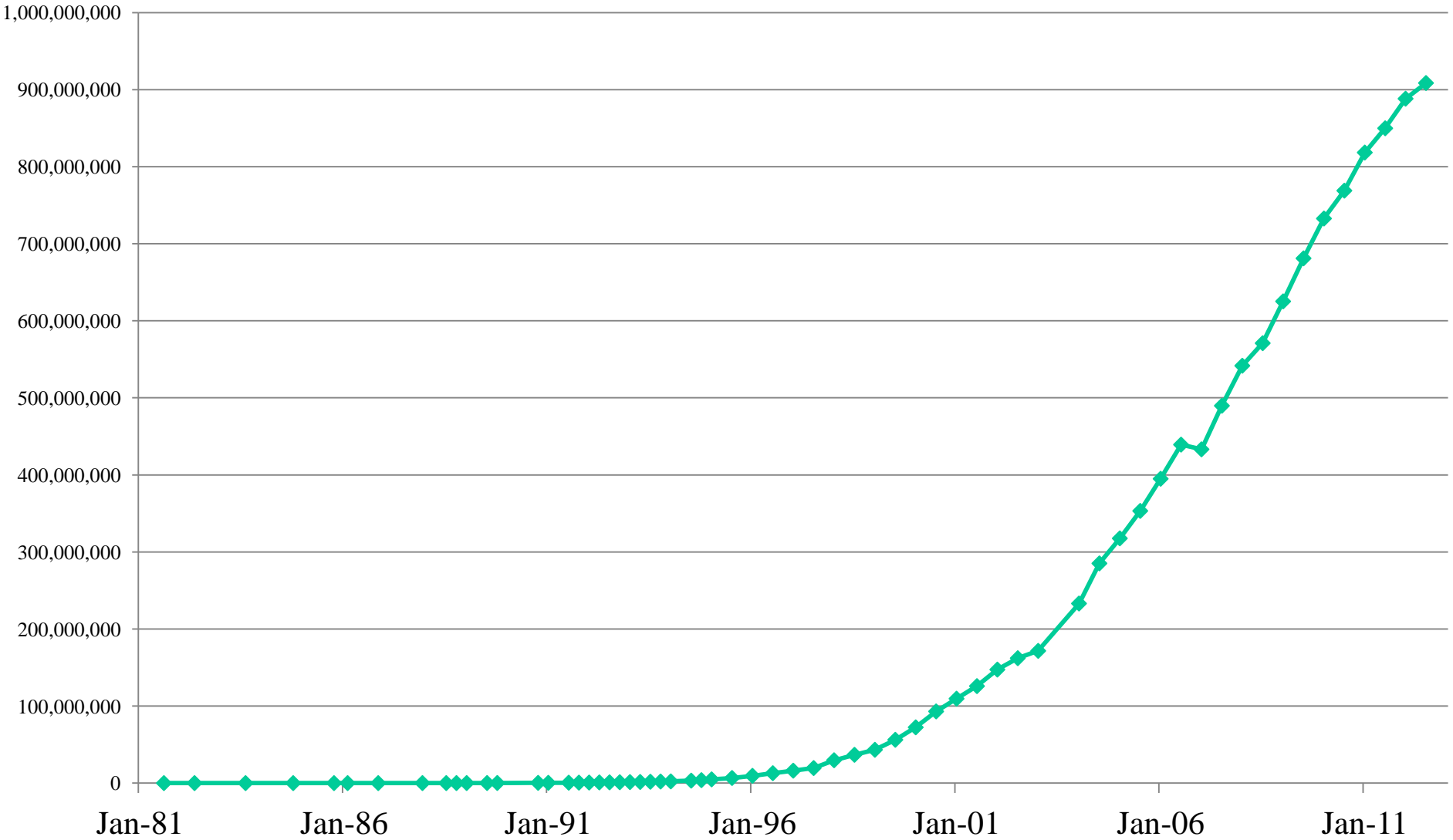
# The Internet

- Global collection of public “IP” networks
  - Private networks are often called “intranets”
- Independent
  - Each organization maintains its own network
- Cooperating
  - Internet Protocol (IP) address blocks
  - Domain names
  - World-Wide Web Consortium (W3C)
  - Computer Emergency Response Team (CERT)

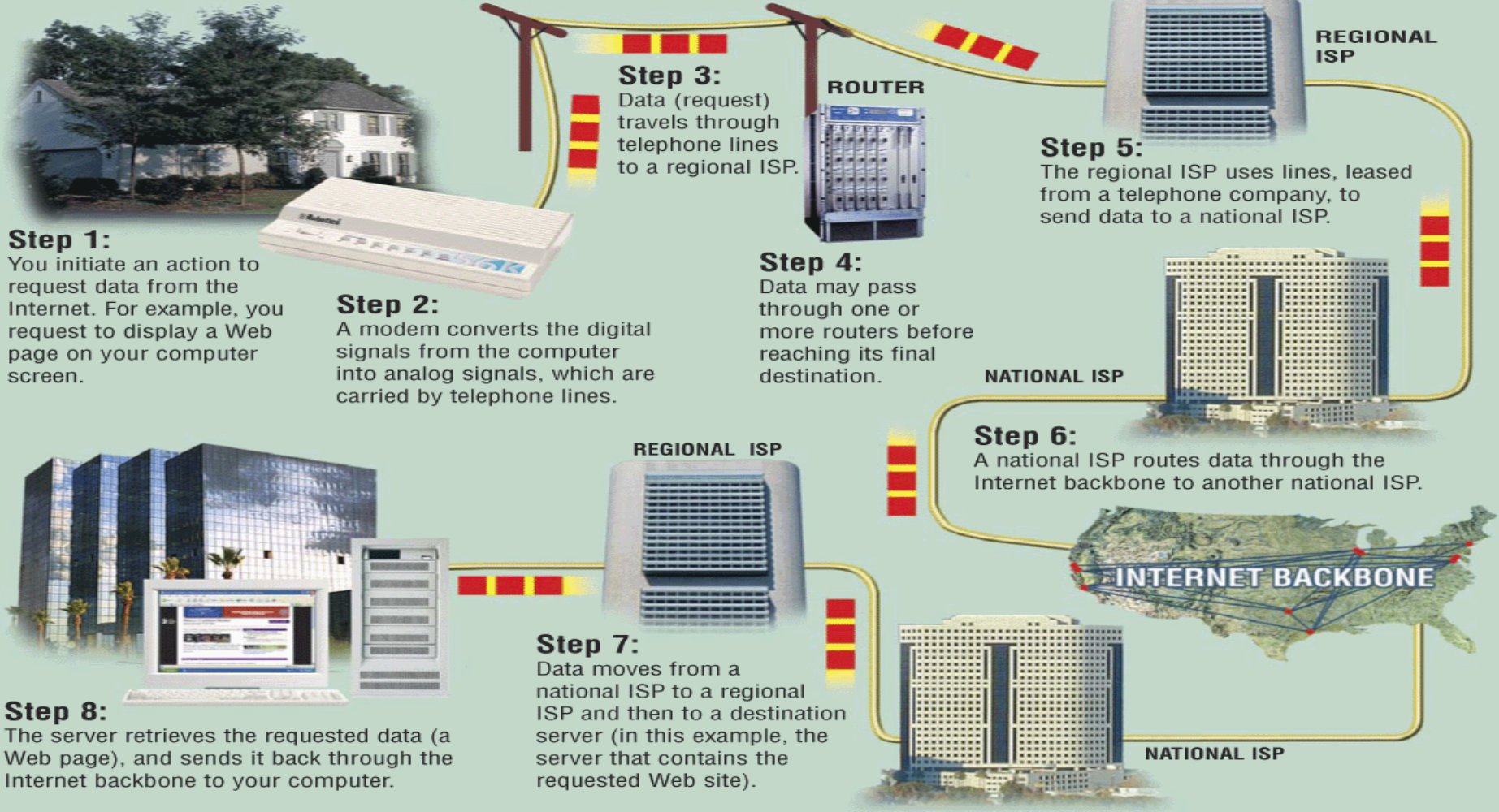
# A Short History of the Internet

- 1969: Origins in government research
  - Advanced Research Projects Agency (ARPAnet)
  - Key standards: UDP, TCP, DNS
- 1983: Design adopted by other agencies
  - Created a need for inter-network connections
  - Key standards: IP
- 1991: World-Wide Web added point-and-click
  - Now 908 million Internet “hosts” (July 2012)
  - Key standards: HTTP, URL, HTML, XML

# What Changed in 1994?



# Overview



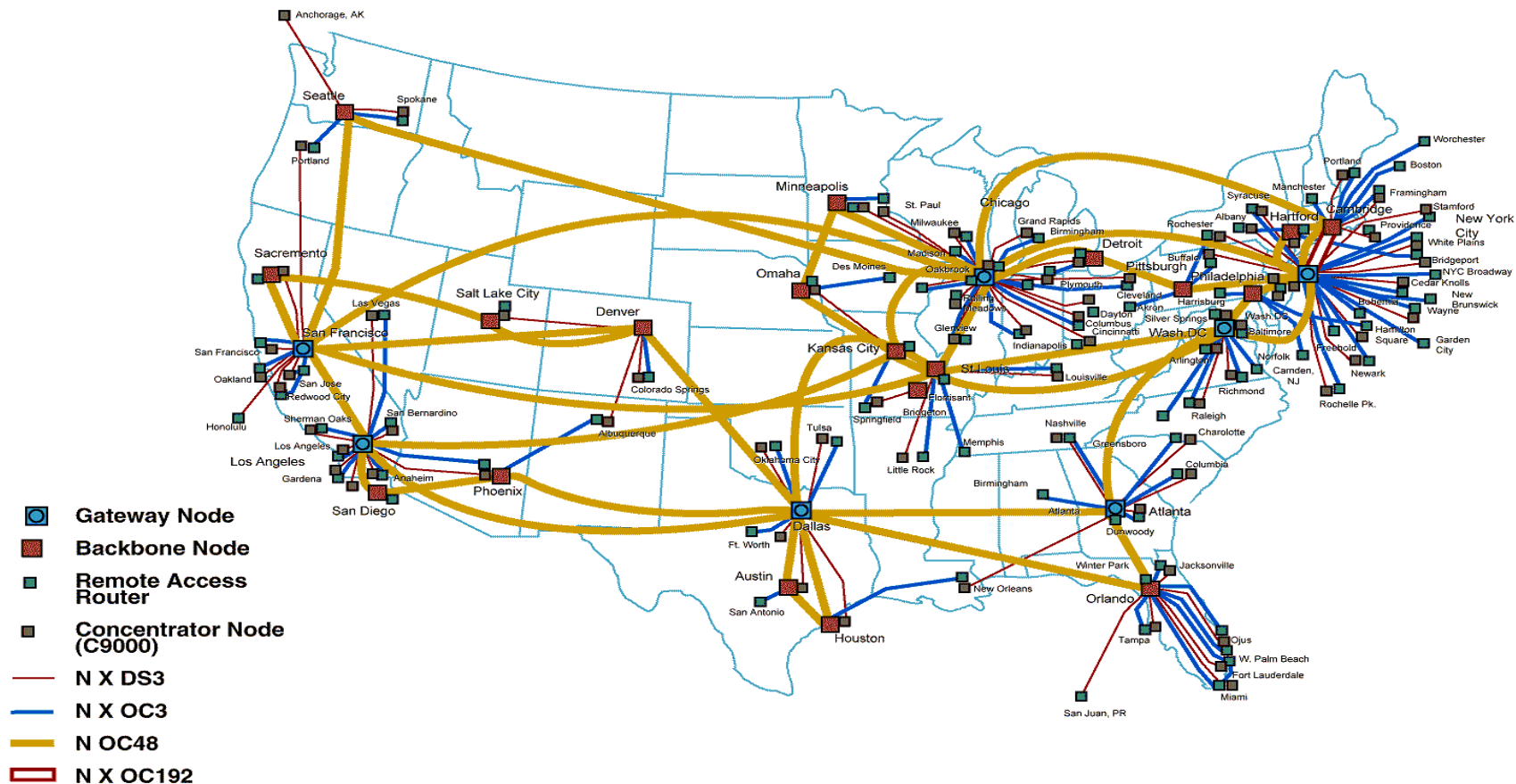
# Types of Digital Channels

- “Backbone”
  - Microwave
  - Satellite
  - Fiber
- “Last mile” wired
  - Telephone modem
  - ADSL
  - Cable modem
  - Fiber
- “Last mile” wireless
  - Wi-Fi (IEEE 802.11)
  - GSM





# AT&T IP BACKBONE NETWORK 2Q2000



Note: map is not to scale.

# Thinking About Speed

- Two parts of moving data from here to there:
  - Getting the first bit there
  - Getting everything there
- Fundamentally, there's no difference:
  - Moving data from the processor to RAM
  - Saving a file to disk
  - Downloading music from a server in China

# Some Definitions

- **Latency**

- The amount of **time** it takes data to travel from source to destination

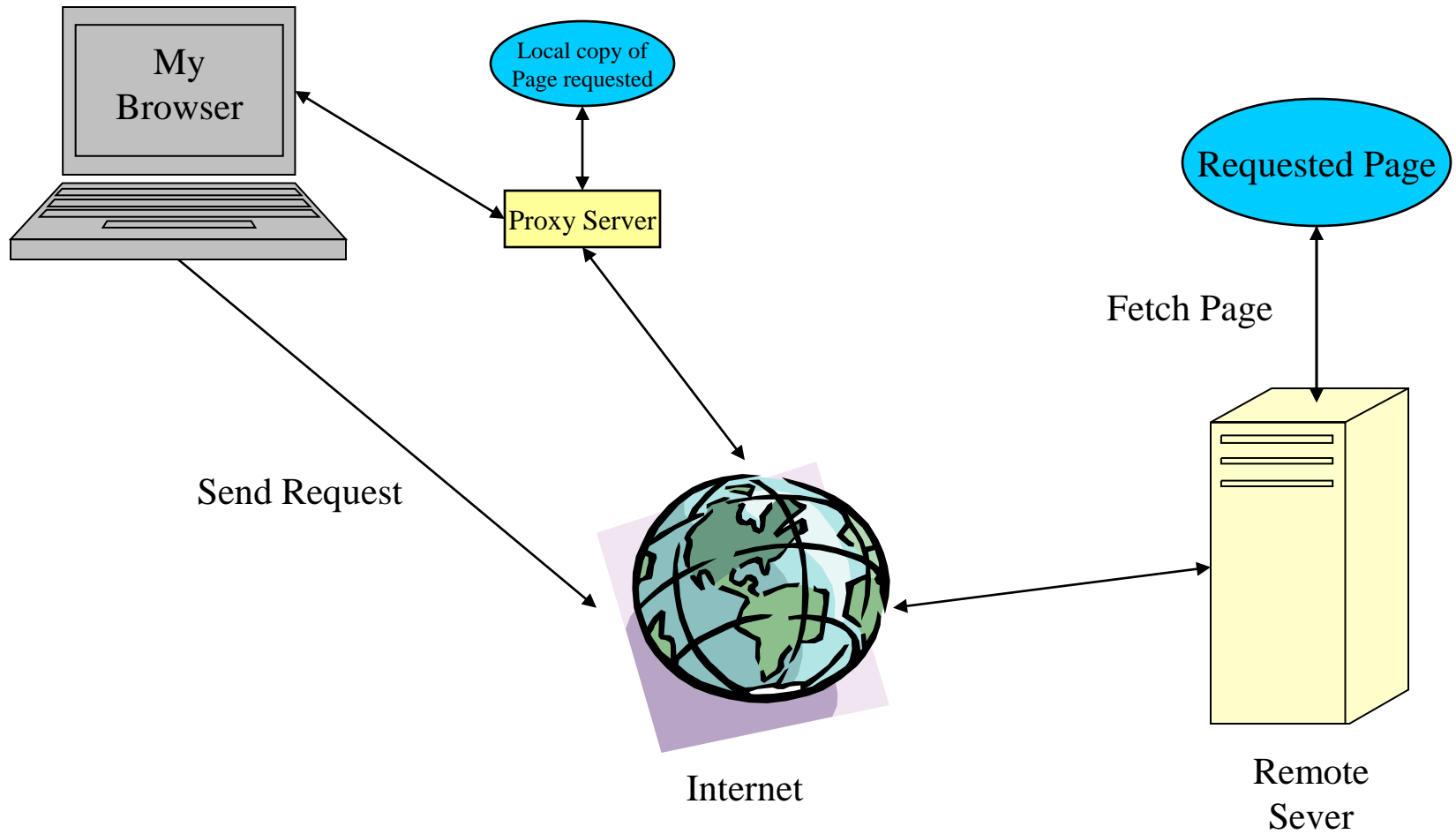
- **Bandwidth**

- The amount of data that can be transmitted in a fixed amount of **time**

# Internet $\neq$ Web

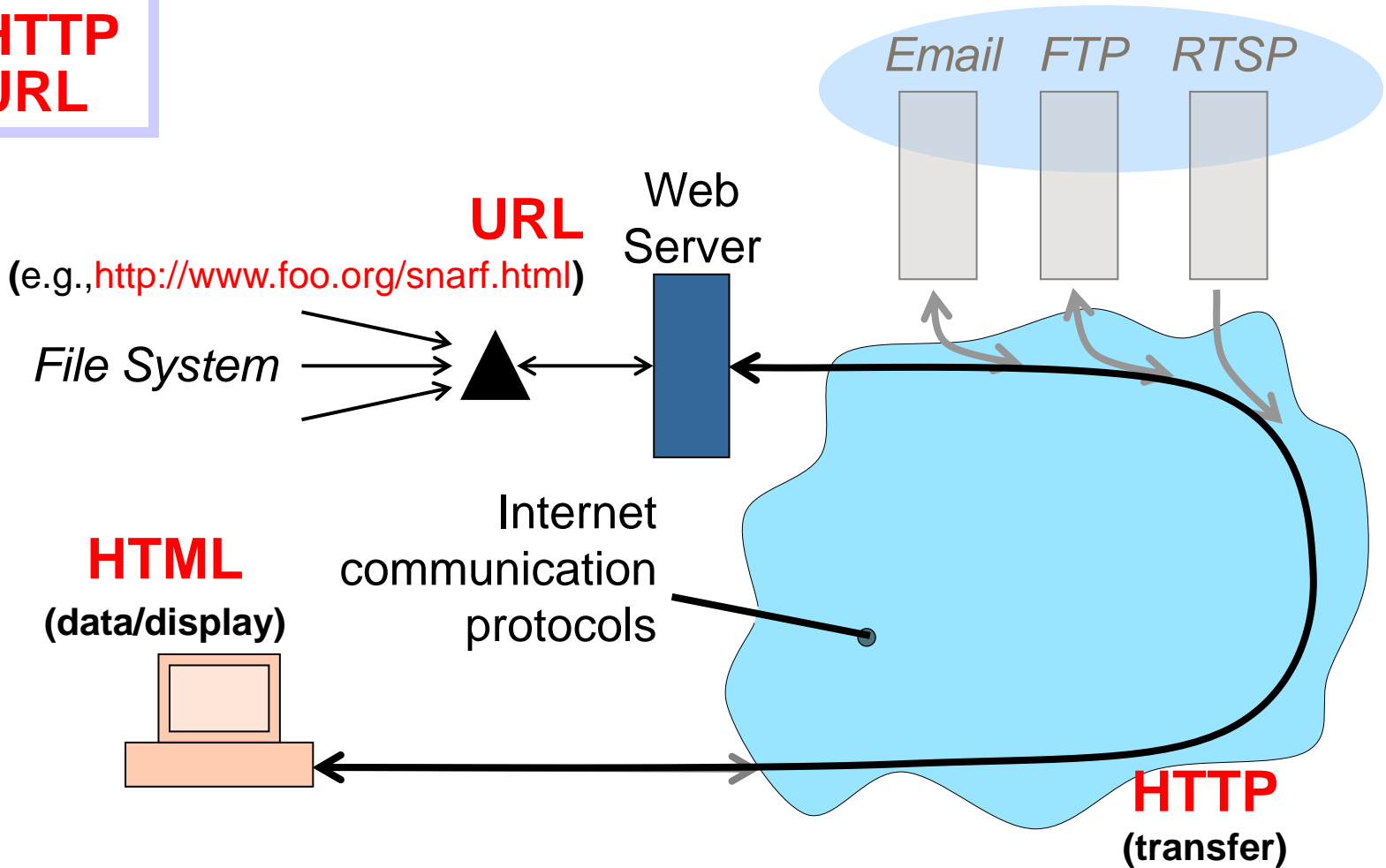
- Internet: collection of global networks
- Web: way of managing information exchange
- There are many other uses for the Internet
  - File transfer (FTP)
  - Email (SMTP, POP, IMAP)

# The World-Wide Web



# “The Web”

**HTML**  
**HTTP**  
**URL**



# Web Standards

- HTML
  - How to write and interpret the information
- URL
  - Where to find it
- HTTP
  - How to get it

# Types of Internet “Nodes”

- Hosts
  - Computers that use the network to do something
- Routers
  - Specialized computers that route packets
- Gateway
  - Routers that connect two networks
- Firewall
  - Gateways that pass packets selectively



# IP Address

- Every host (and every router) is identified by an “Internet Protocol” (IP) address
- 32 bit number, divided into four “octets”

128.8.11.33

216.239.39.99

199.181.132.250

**Example: point your browser at “<http://66.249.93.99/>”**

# An Internet Protocol (IP) Address

Identifies a LAN

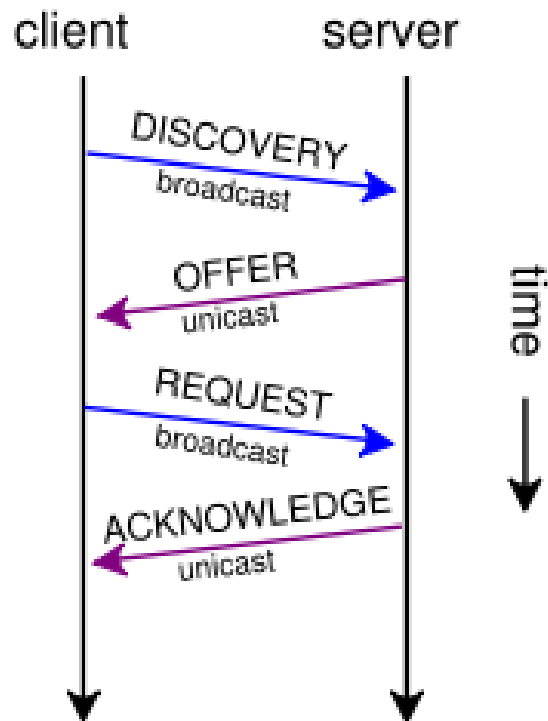
IP address:

216.183.103.150

Identifies a specific computer

# Dynamic IP Addresses

- Dynamic Host Configuration Protocol (DHCP)



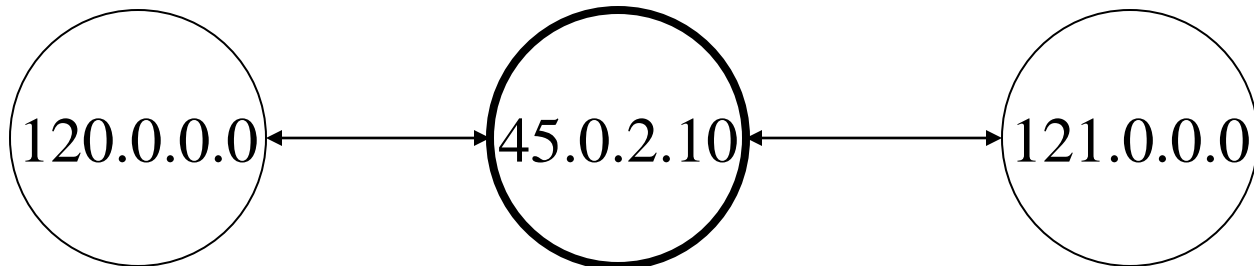
# Hands-on:

## Learn About Your IP Address

- Find your IP address
  - Bring up a command window
    - In Windows, type “cmd” in the search box!
  - Type “ipconfig /all” (and press enter)
- See who “owns” that address
  - Use <http://remote.12dt.com/>
- See where in the world it (probably) is
  - <http://www.geobytes.com/ipLocator.htm>

# Routing Tables

<b>IP Prefix</b>	<b>Next Router</b>	<b>Estimated Delay</b>
216.141.xxx.xxx	120.0.0.0	18 ms
216.xxx.xxx.xxx	121.0.0.0	34 ms
101.42.224.xxx	120.0.0.0	21 ms
xxx.xxx.xxx.xxx	121.0.0.0	250 ms



# TraceRoute

- See how packets get from South Africa to you
  - <http://services.truteq.com/>
- Look at the same data visually
  - <http://visualroute.visualware.com/>

# Domain Name Service (DNS)

- “Domain names” improve usability
  - Easier to remember than IP addresses
  - Written like a postal address: specific-to-general
- Each “name server” knows one level of names
  - “Top level” name servers know .edu, .com, .mil, ...
  - .edu name server knows umd, umbc, stanford, ...
  - .umd.edu name server knows wam, ischool, ttclass, ...
  - .wam.umd.edu name server knows rac1, rac2, ...

# IP Addresses and Domain Names

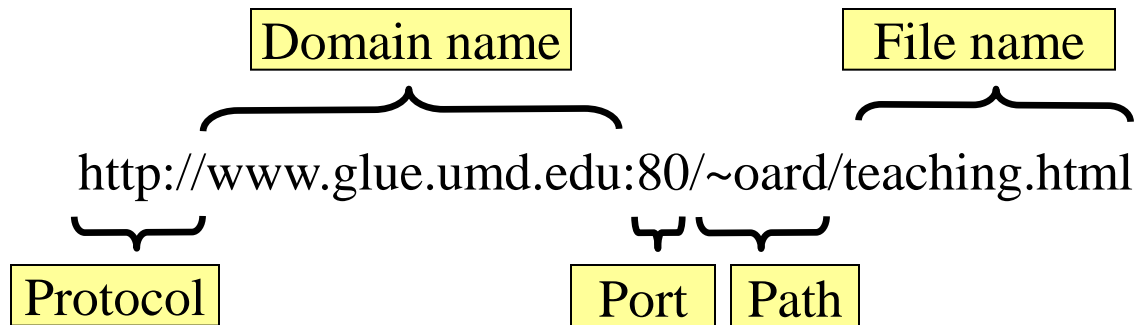
IP address: 128.8.10.142

Domain Name: wam.umd.edu



# Uniform Resource Locator (URL)

- Uniquely identify Web pages



# Ports

- Well-known ports
  - 22 Secure Shell (for SSH and SFTP)
  - 25 Simple Mail Transfer Protocol (SMTP)
  - 53 Domain Name System (DNS)
  - 68 Dynamic Host Configuration Protocol (DHCP)
  - 80 Hypertext Transfer Protocol (HTTP)
  - 143 Internet Message Access Protocol (IMAP)
  - 554 Real-Time Streaming Protocol (RTSP)
- Registered Ports
  - 8080 HTTP server run by ordinary users
- Ephemeral Ports

# Port Mapping

- Internet Service providers lease one IP address
  - But home networks may contain many machines
- Network Address Translation (NAT)
  - Each internal machine gets a private IP address
  - Ports on internal machines are mapped both ways
- Port forwarding
  - Permits public server to run in the local network

# Paths

- Specify location of files on a hard drive
- Folder metaphor
  - Hierarchically nested directories
    - `/afs/wam.umd.edu/home/wam/j/i/jimmylin/home`
    - `C:\Documents and Settings\Jimmy Lin\My Documents`
  - Absolute vs. relative paths
    - `../pub`
    - `..\Desktop`
    - `~/oard`

# Hands On:

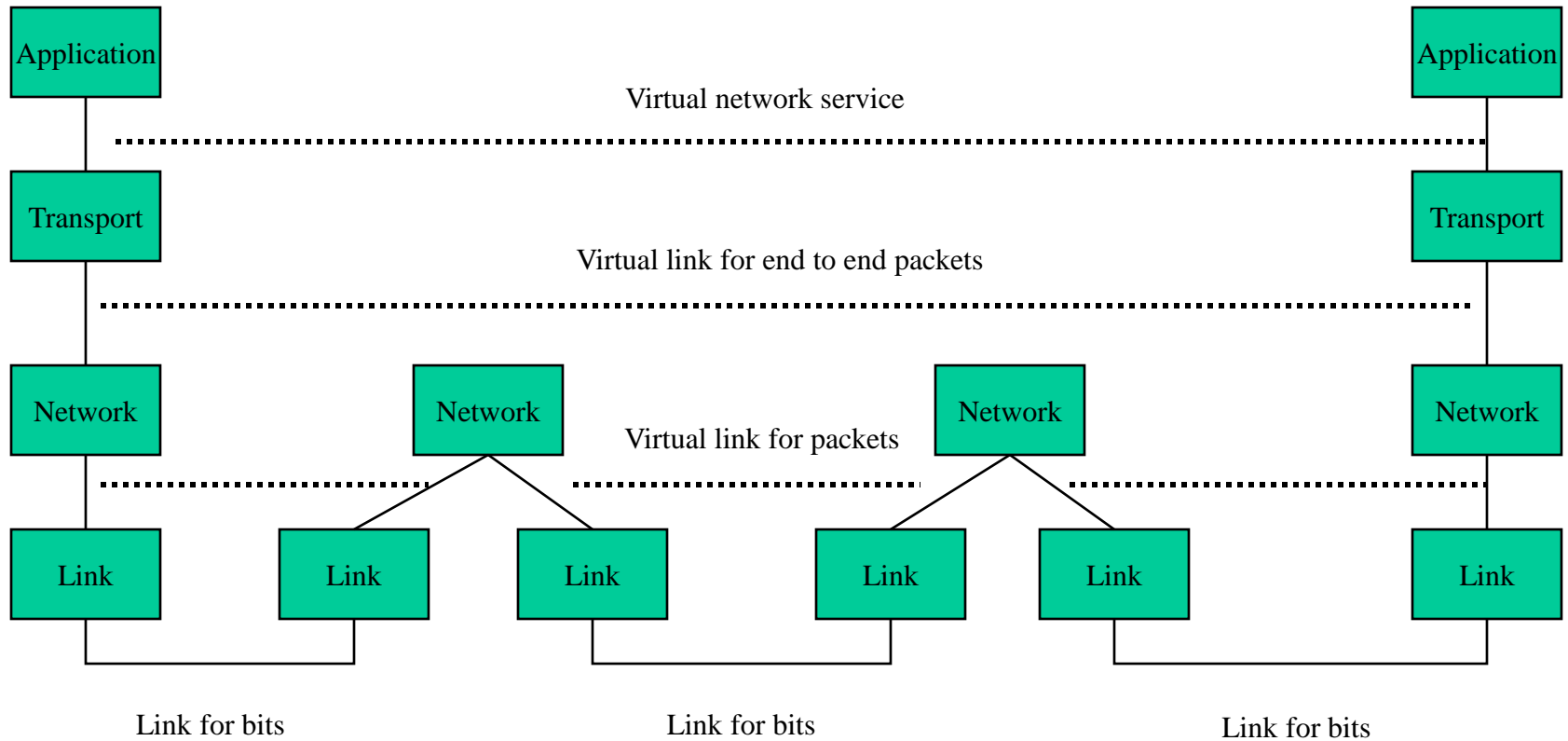
## The Directory Tree

- First, use Windows Explorer to visually explore the directory tree
- Now launch a “shell” (type cmd in search box)
  - “c:” takes you to Drive C
  - “dir” lists the present “directory”
  - “cd WINDOWS” takes you “down” to the WINDOWS directory
  - cd .. takes you “up” in the tree

# The TCP/IP “Protocol Stack”

- Link layer moves bits
  - Ethernet, cable modem, DSL
- Network layer moves packets
  - **IP**
- Transport layer provides services to applications
  - UDP, **TCP**
- Application layer uses those services
  - DNS, SFTP, SSH, ...

# TCP/IP layer architecture



# Transmission Control Protocol (TCP)

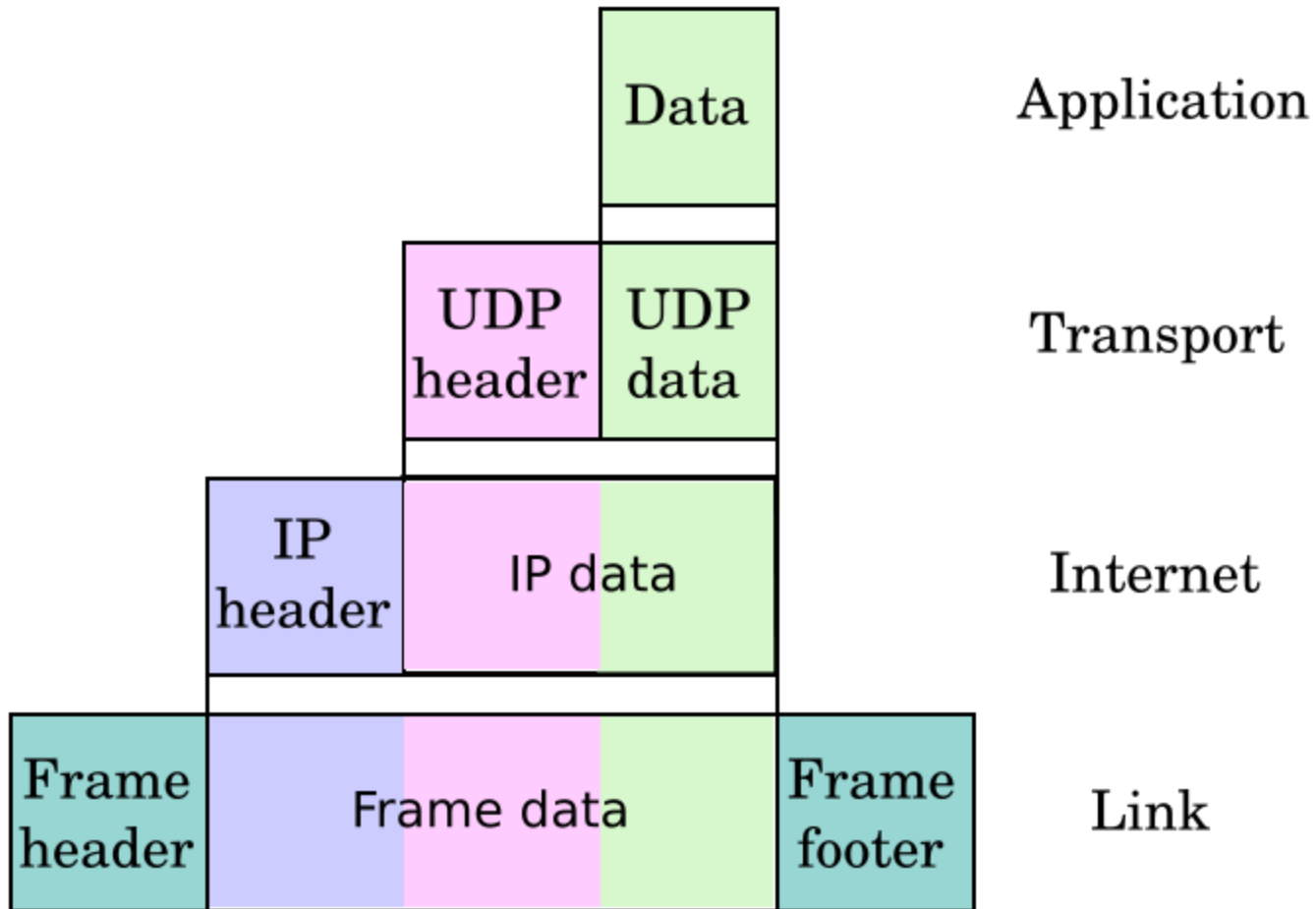
- Built on the network-layer version of UDP
- Guarantees delivery all data
  - Retransmits missing data
- Guarantees data will be delivered in order
  - “Buffers” subsequent packets if necessary
- No guarantee of delivery time
  - Long delays may occur without warning



# User Datagram Protocol (UDP)

- The Internet's basic transport service
  - Sends every packet immediately
  - Passes received packets to the application
- No delivery guarantee
  - Collisions can result in packet loss
- Example: sending clicks on web browser

# UDP/IP Protocol Stack



# HyperText Transfer Protocol (HTTP)

- Send request

GET /path/file.html HTTP/1.0

From: someuser@jmarshall.com

User-Agent: HTTPTool/1.0

- Server response

HTTP/1.0 200 OK

Date: Fri, 31 Dec 1999 23:59:59 GMT

Content-Type: text/html

Content-Length: 1354

<html><body> <h1>Happy New Millennium!</h1> ... </body> </html>

# File Transfer Program (FTP)

- Used to move files between machines
  - Upload (put) moves from client to server
  - Download (get) moves files from server to client
- Both visual and command line interfaces available
- Normally requires an account on the server
  - Userid “anonymous” provides public access

# Hands On: Graphical Secure FTP

- SFTP to “terpconnect.umd.edu”
- Change directory to “/pub/**USERID**”
- Upload or download files
- You can see these files at:  
[http://terpconnect.umd.edu/~\*\*USERID\*\*/](http://terpconnect.umd.edu/~<b>USERID</b>/)

# Network Abuse

- Flooding
  - Excessive activity, intended to prevent valid activity
- Worms
  - Like a virus, but self-propagating
- Sniffing
  - Monitoring network traffic (e.g., for passwords)

# Encryption

- Secret-key systems (e.g., DES)
  - Use the same key to encrypt and decrypt
- Public-key systems (e.g., PGP)
  - Public key: open, for encryption
  - Private key: secret, for decryption
- Digital signatures
  - Encrypt with private key, decrypt with public key

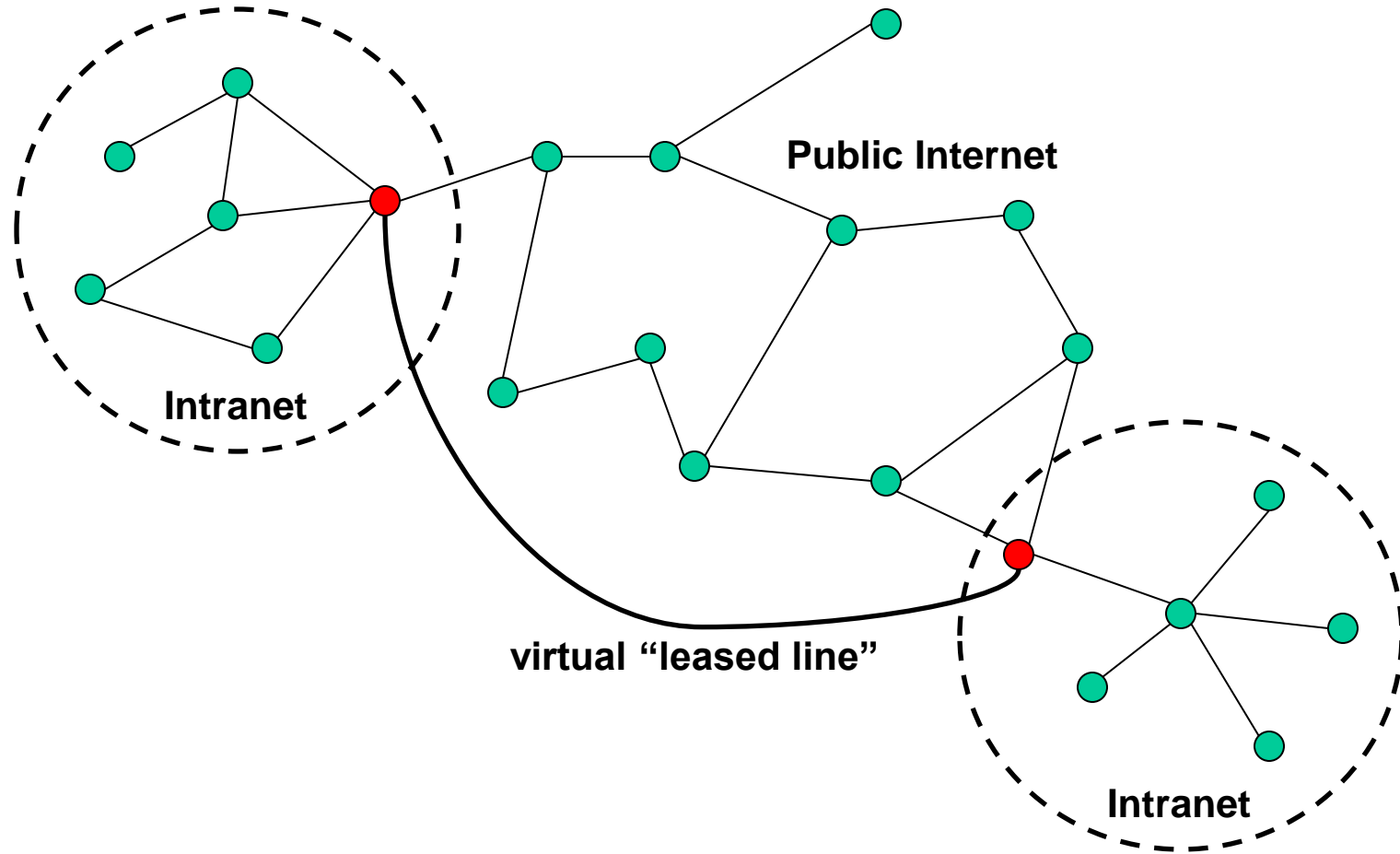
# Encrypted Standards

- Secure Shell (SSH)
  - Replaces Telnet
- Secure FTP (SFTP)/Secure Copy (SCP)
  - Replaces FTP
- Secure HTTP (HTTPS)
  - Used for financial and other private data
- Wired Equivalent Protocol (WEP)
  - Used on wireless networks
- Virtual Private Network (VPN)
  - Not really a “standard”



# Virtual Private Networks

a secure private network over the public Internet



# HyperText Markup Language (HTML)

- Simple document structure language for Web
- Advantages
  - Adapts easily to different display capabilities
  - Widely available display software (browsers)
- Disadvantages
  - Does not directly control layout

# “Hello World” HTML This is the header

```
<html>  
<head>  
<title>Hello World!</title>  
</head>
```

```
<body>  
  
<p>Hello world! This is my first webpage!</p>  
  
</body>  
</html>
```

This is the actual content of the HTML document

# Hands On:

## Learning HTML From Examples

- Use Internet Explorer to find a page you like
  - <http://www.glue.umd.edu/~oard>
- On the “Page” menu select “View Source” (in IE7)
  - Opens a notepad window with the source
- Compare HTML source with the Web page
  - Observe how each effect is achieved

# Hands On: “Adopt” a Web Page

- Modify the HTML source using notepad
  - For example, change the page to yours
- Save the HTML source somewhere
  - In the “File” menu, select “Save As”
  - **Put the name in quotes (e.g., “test.html”)**
- FTP it to your ../pub directory on terpconnect
- View it
  - [http://www.wam.umd.edu/~\(yourlogin\)/test.html](http://www.wam.umd.edu/~(yourlogin)/test.html)

# Tips

- Edit files on your own machine
  - Upload when you're happy
- Save early, save often, just save!
- Reload browser to see changes
- File naming
  - Don't use spaces
  - Punctuation matters

# HTML Document Structure

- “Tags” mark structure
  - `<html>a document</html>`
  - `<ol>an ordered list</ol>`
  - `<i>something in italics</i>`
- Tag name in angle brackets `<>`
  - Not case sensitive
- Open/Close pairs
  - Close tag is sometimes optional (if unambiguous)

# Logical Structure Tags

- Head
  - Title
- Body
  - Headers: `<h1>` `<h2>` `<h3>` `<h4>` `<h5>`
  - Lists: `<ol>`, `<ul>` (can be nested)
  - Paragraphs: `<p>`
  - Definitions: `<dt>``<dd>`
  - Tables: `<table>` `<tr>` `<td>` `</td>` `</tr>` `</table>`
  - Role: `<cite>`, `<address>`, `<strong>`, ...



# Physical Structure Tags

- Font

- Typeface: `<font face="Arial"></font>`

- Size: `<font size="+1"></font>`

- Color: `<font color="990000"></font>`

- [http://webmonkey.wired.com/webmonkey/reference/color\\_codes/Emphasis](http://webmonkey.wired.com/webmonkey/reference/color_codes/Emphasis)

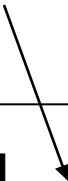
- Bold: `<b></b>`

- Italics: `<i></i>`

# (Hyper)Links

## index.html

```
<html>
<head>
<title>Hello World!</title>
</head>
<body>
<p>Hello world! This is my first webpage!</p>
<p>Click <a href="test.html">here</a> for another page.</p>
</body>
</html>
```



## test.html

```
<html>
<head>
<title>Another page</title>
</head>
<body>
<p>This is another page.</p>
</body>
</html>
```

# Hypertext “Anchors”

- Internal anchors: somewhere on the same page
  - `<a href="#students"> Students</a>`
    - Links to: `<a name="students">Student Information</a>`
- External anchors: to another page
  - `<a href="http://www.clis.umd.edu">CLIS</a>`
  - `<a href="http://www.clis.umd.edu#students">CLIS students</a>`
- URL may be complete, or relative to current page
  - `<a href="video/week2.rm">2</a>`
- File name part of URL is case sensitive (on Unix servers)
  - Protocol and domain name are not case sensitive

# Images

- `` *or* ``
  - ``
  - SRC: can be url or path/file
  - ALT: a text string
  - ALIGN: position of the image
  - WIDTH and HEIGHT: size of the image
- Can use as anchor:
  - `<a href=URL><img src=URL2></a>`
- Example:
  - <http://www.umiacs.umd.edu/~daqingd/Image-Alignment.html>

# Tables

eenie	mennie	miney
mo	catch	a tiger
by	the	toe

# Table Example

```
<table align="center">
```

```
<caption align="right">The caption</caption>
```

```
< tr align="LEFT">
```

```
  <th> Header1 </th>
```

```
  <th> Header2</th>
```

```
</tr>
```

```
<tr><td>first row, first item </td>
```

```
  <td>first row, second item</td></tr>
```

```
< tr><td>second row, first item</td>
```

```
  <td>second row, second item</td></tr>
```

```
</table>
```

See also: <http://www.umiacs.umd.edu/~daqingd/Simple-Table.html>

# Before You Go

On a sheet of paper, answer the following (ungraded) question (no names, please):

What was the muddiest point in today's class?