

Web Networking, The Internet, HTML/CSS

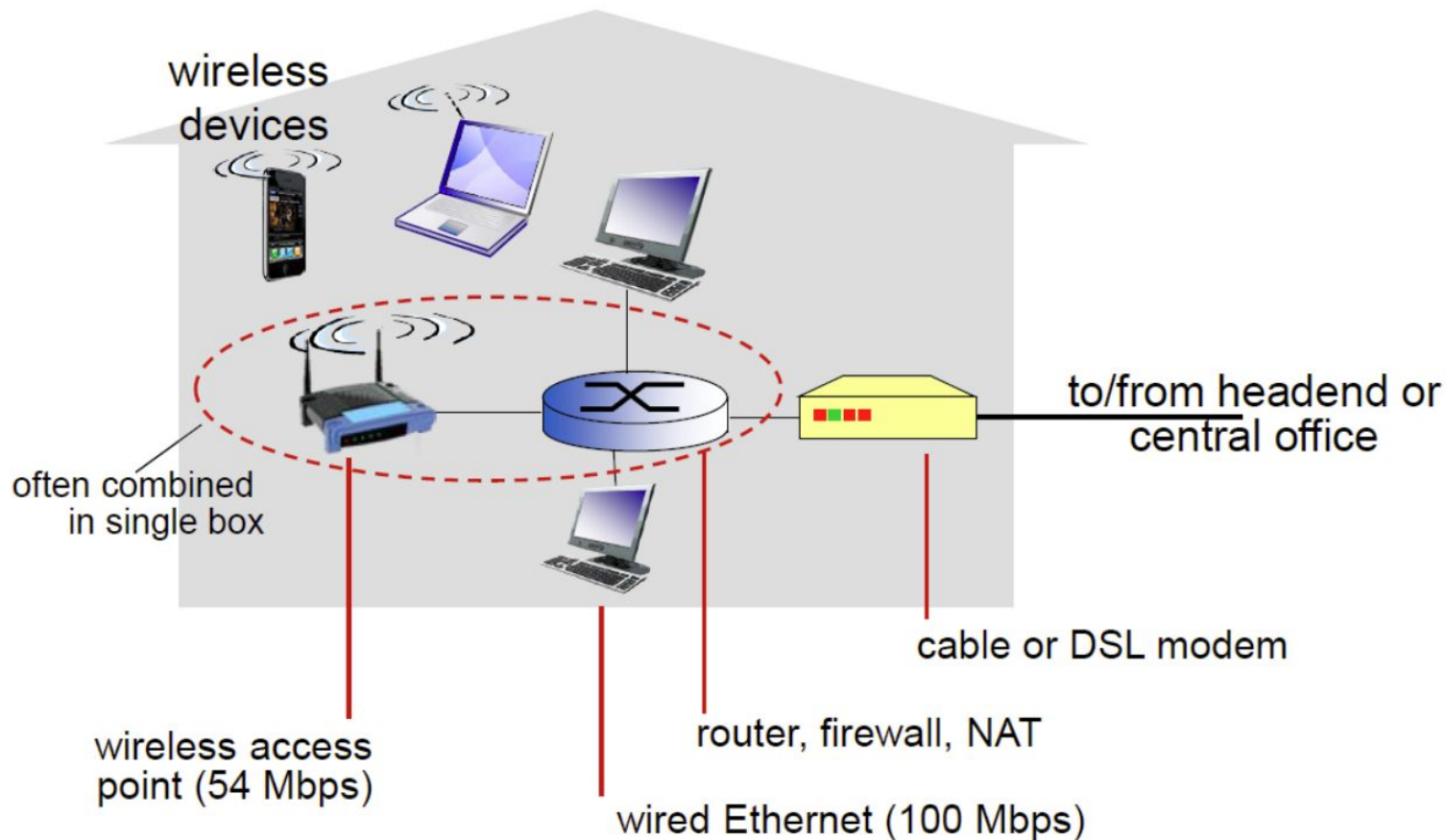


CS302 2019
Hammond Pearce

Networking

- Exchange of data between computers
- Any device that
 - Originates / sends
 - Routes / transfers
 - Terminates / receives
- Networks and the internet are built on the *routing* of data
- Clients make requests of servers
 - Servers can also make requests of clients

Home Network - connected to the Internet?



Networking

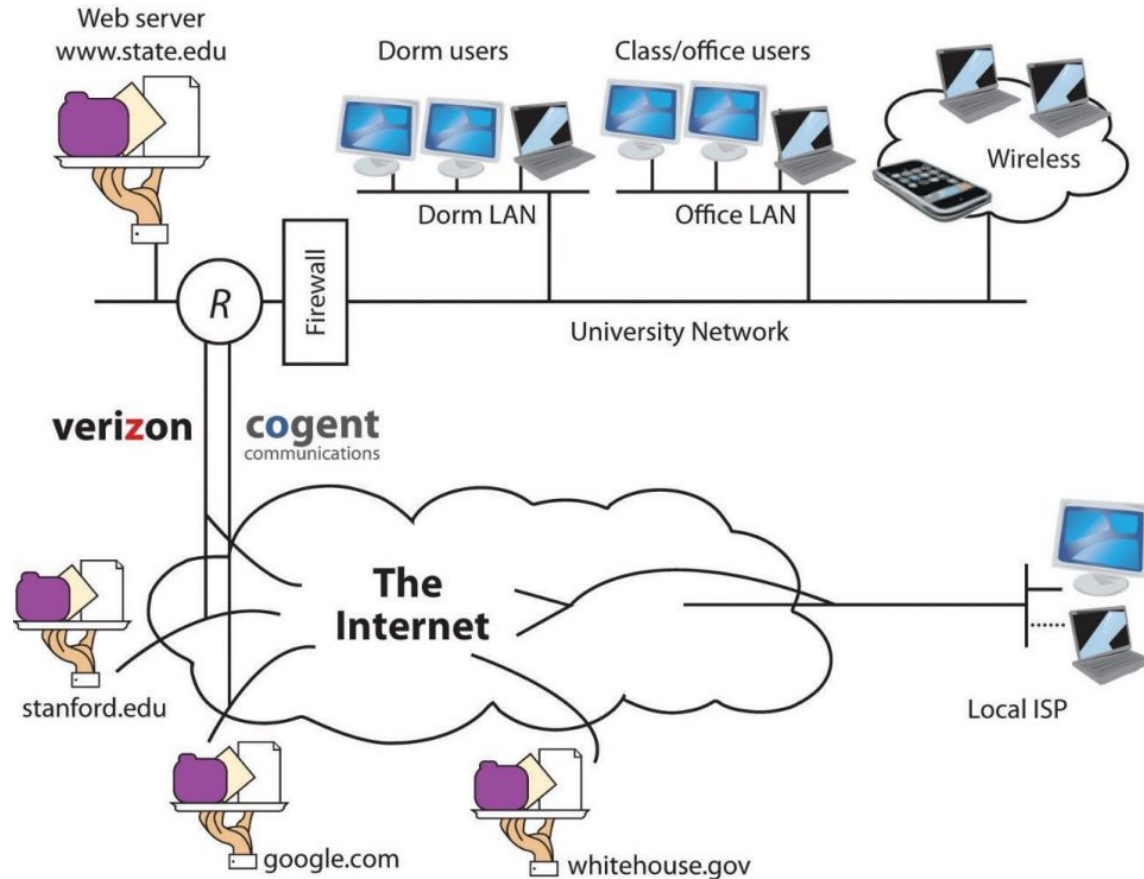


Image from *Getting the most out of information systems, v1.4*

Internet backbones

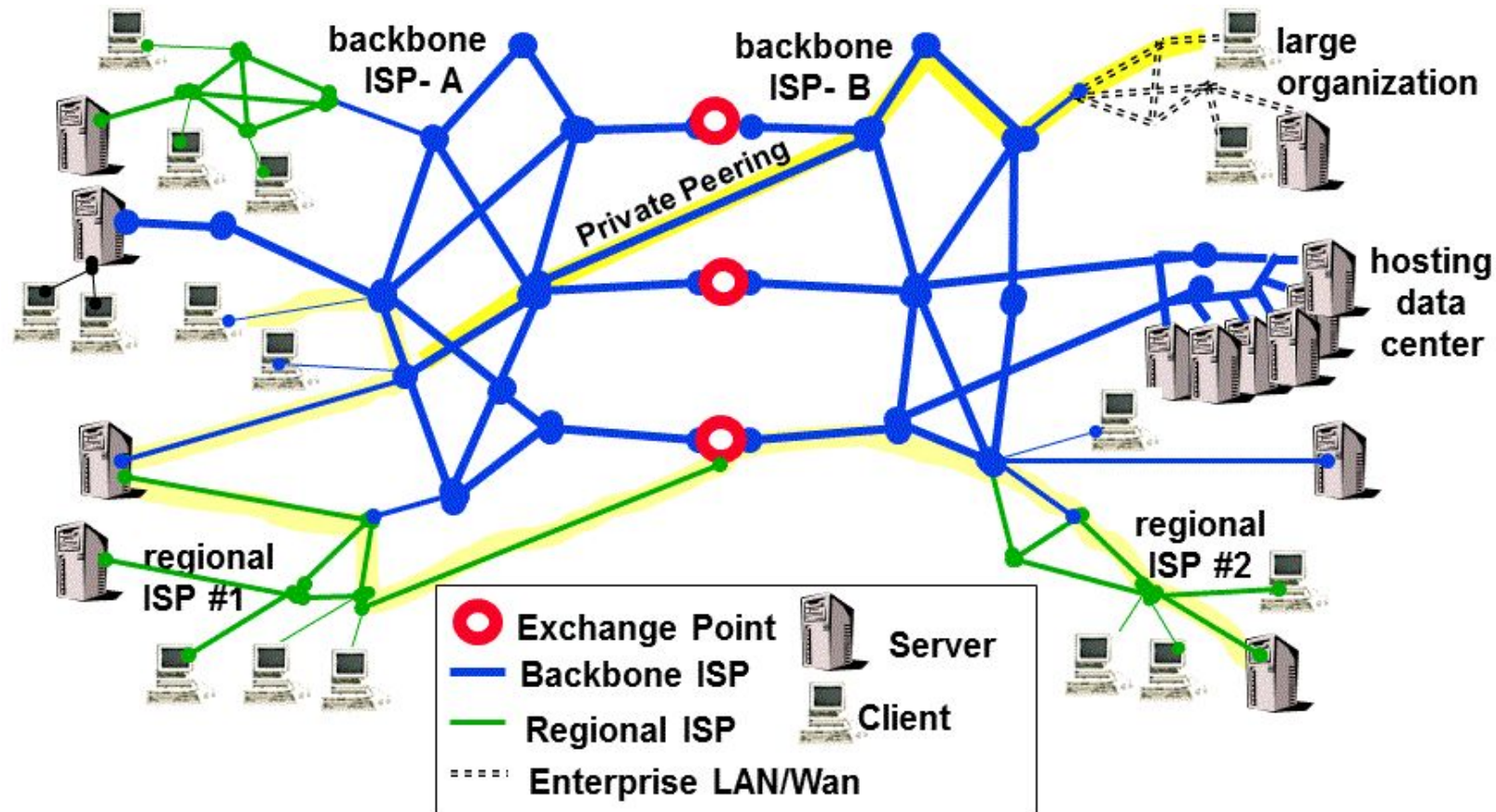


Image from <http://navigators.com/sessphys.html>

Interactive demo

Open <https://www.submarinecablemap.com/>

And that's just the undersea cables!

- Land-based cables
- Microwave links
- Satellite links

Demo: traceroute

Demo: ping

Networking Protocols

- In order for any communication to happen, both parties must understand
 - E.g. I speak English, you understand English
- Computer communication is the same
 - A string of 0's and 1's must be understandable
- You've used protocols before
 - E.g. UART

Networking Protocols

- Networking protocols describe digital communication
 - Data is discretized into packets
 - A packet has a *header*, *data*, and *footer*
 - Header describes source and destination
- Computers use *ports* as gates to send and receive data
- Internet uses IP addresses to identify connections e.g. 130.216.10.24
- We combine e.g. 130.216.10.24:80
 - Address:Port is like Street Address:Flat no.

Internet Protocols

OSI Model	Internet Model	Internet Protocols
Application	Application	HTTP, HTTPS, SSH, DNS, SSL, FTP, POP3, SMTP, IMAP, Telnet, NNTP
Presentation		
Session		
Transport	Transport	TCP, UDP
Network	Network	IP, ICMP, ARP, DHCP
Datalink	Network Link	Ethernet, PPP, ADSL
Physical		

Basic Internet Networking Example

- Client requests “www.google.com” in browser
- Browser infers “https” connection method, default port is 443
- Google.com IP address loaded
- Request sent
- Data returns, formatted to https
- This encodes HTML, which is decrypted and displayed to user
- Beyond what is “seen”, there are also a lot of headers and control messages

Demo: Chrome/Firefox networking tools

HTML, CSS, and Javascript

- “Hyper Text Markup Language”
 - Markup languages describe formatting of text
 - E.g. LaTeX, HTML, XHTML, XML, Markdown
- HTML is the markup language of the internet
- Lots of online tutorials
- Now up to HTML5
- *Rarely used in isolation.*
 - CSS for styling
 - Javascript for interactivity



HTML crash course

- Tags using `<` and `>`
 - `<open>` and `</close>`
- Examples
 - `This is bold text `
 - `This is a link to google`
- Tags can be nested
 - `<i>This is both bold and italic</i>`
- Pages are made up of
 - `<html>`
 - `<head>...</head>`
 - `<body>...</body>`
 - `</html>`

- You can make HTML in any text editor
 - E.g. Notepad++, gedit, vim, Visual Studio Code
- You can also make it in WYSIWYGs
 - E.g. Dreamweaver, Frontpage, Office
- We won't go through every tag
 - There's quite a few!
 - You can see what other websites have used using *Inspect Element* (in browser)
- Instead, let's have a demo

In-class demo (05.1)

Make a simple HTML page with a `<head><title>` and

`<body><h1><p>` paragraphs and

``bolded text and

`` red text

- Typically websites don't style their HTML directly anymore
- Cascaded Style Sheets provide a decoupled-flexible mechanism instead
 - Markup language for styling HTML
- For each tag, specify attributes

```
body {  
    color: purple; /* font color */  
    background-color: green;  
}
```

- There are a *lot* of possible attributes

- CSS is based around selectors and declarations
- In the previous example,
 - selector is “body”,
 - declarations are property: value
- The styles *cascade*
 - A given tag may be affected by multiple styles simultaneously

```
body {  
  color: purple; /* font color */  
}  
p {  
  Background-color: green;  
}
```

```
<body>  
  <p>This text will be purple, and  
    it will have a green background  
  </p>  
</body>
```


Link CSS to HTML

- HTML for content, CSS for style
- CSS can be embedded into `<style>` tag `</style>`
- It can also be embedded onto tags
 - `<p style="color:red">This is red</p>`
- However, best to use global CSS files for consistency and efficiency
- In `<head>`,
 - `<link href="stylesheet.css" rel="stylesheet" type="text/css" />`
- Use classes to define subtypes of tags
 - `<div class="tweet">HeaderTweet contents</div>`

In-class demo (05.2)

Modify HTML from 05.1 to call external CSS file

Set some attributes

Web design tips

- Modular design will make your life easier
- Use stylesheets across pages for consistency
- Simple designs are the best designs
- 80-20 rule: 20% of website = 80% of useful content
- Make use of whitespace to guide the user
- Majority of people don't scroll down
- Limit your colour scheme to just a few tones
- Think about compatibility! TEST!
- <http://www.theworldsworstwebsiteever.com/>

What about dynamic content?

- Websites don't tend to be static
- Content will change over time/sessions
- On the server side, we use webserver to dynamically create HTML
 - and sent it to clients
- Python/CherryPy can be used to build a webserver
 - Load HTML and CSS files
 - Modify them based on databases, computation
- We could also use other languages/frameworks
 - RubyOnRails, Go, ASP.net ...
- *(We can also create dynamic content locally using JS. This is for later...)*

Conclusions

- A variety of networking protocols are used in reality
- For the internet, we typically serve HTML/CSS/JS over TCP/IP to an IP:PORT
- A variety of HTML and CSS tags and attributes exist
- Lots of online resources
- We can dynamically create HTML