Networking and the Web

Day 1, Week 2 - WDI NYC June 2013

Internet Protocol

- Primary communications protocol for relaying data across network boundaries
- Principle protocol of the *Internet Protocol Suite*
- Delivers packets from the source host to the destination host solely based on the *IP Address*
- IPv4 (first major version) is the dominant protocol of the internet
- IPv6 is the next major version, which allows for a different address

https://en.wikipedia.org/wiki/Internet_Protocol

IP Address

- Internet Protocol Address
- Example: 198.228.235.120 (IPv4) or 2607:f0d0:1002:0051:0000:0000:0000:0004 (IPv6)
- Every system on internet has one
- Not fixed to each system- they can change
- Systems with an IP address, connected to the internet, can connect to each other

https://en.wikipedia.org/wiki/IP_address

TCP

- Transmission Control Protocol
- One of the original core protocols of the *Internet Protocol Suite*
- Provides reliable, ordered, error-checked delivery of a stream of octets between programs running on computers
- Web Browsers use TCP when they connect to servers on the World Wide Web
- Also used to transfer email and files

https://en.wikipedia.org/wiki/Transmission_Control_Protocol

UDP

- User Datagram Protocol
- One of the original core protocols of the *Internet Protocol Suite*
- *Not* used by Web Browsers for web pages. More for things like Skype.
- Simple transmission model with a minimum of protocol mechanism
- Faster, less reliable, fewer features

https://en.wikipedia.org/wiki/User_Datagram_Protocol

TCP v UDP

TCP	UDP
Reliable	Unreliable
Ordered	Not Ordered
Heavyweight	Lightweight
Streaming	Datagrams
Bidirectional	No congestion control

https://en.wikipedia.org/wiki/User_Datagram_Protocol#Comparison_of_UDP_and_TCP

Packets

- Formatted unit of data carried by packet mode computer network (internet)
- Without packets, raw bits, bytes or characters are transmitted on the network
- Contains two kinds of data: control information and payload
- On internet, control information tells data its destination

http://en.wikipedia.org/wiki/Network_packet

Ports

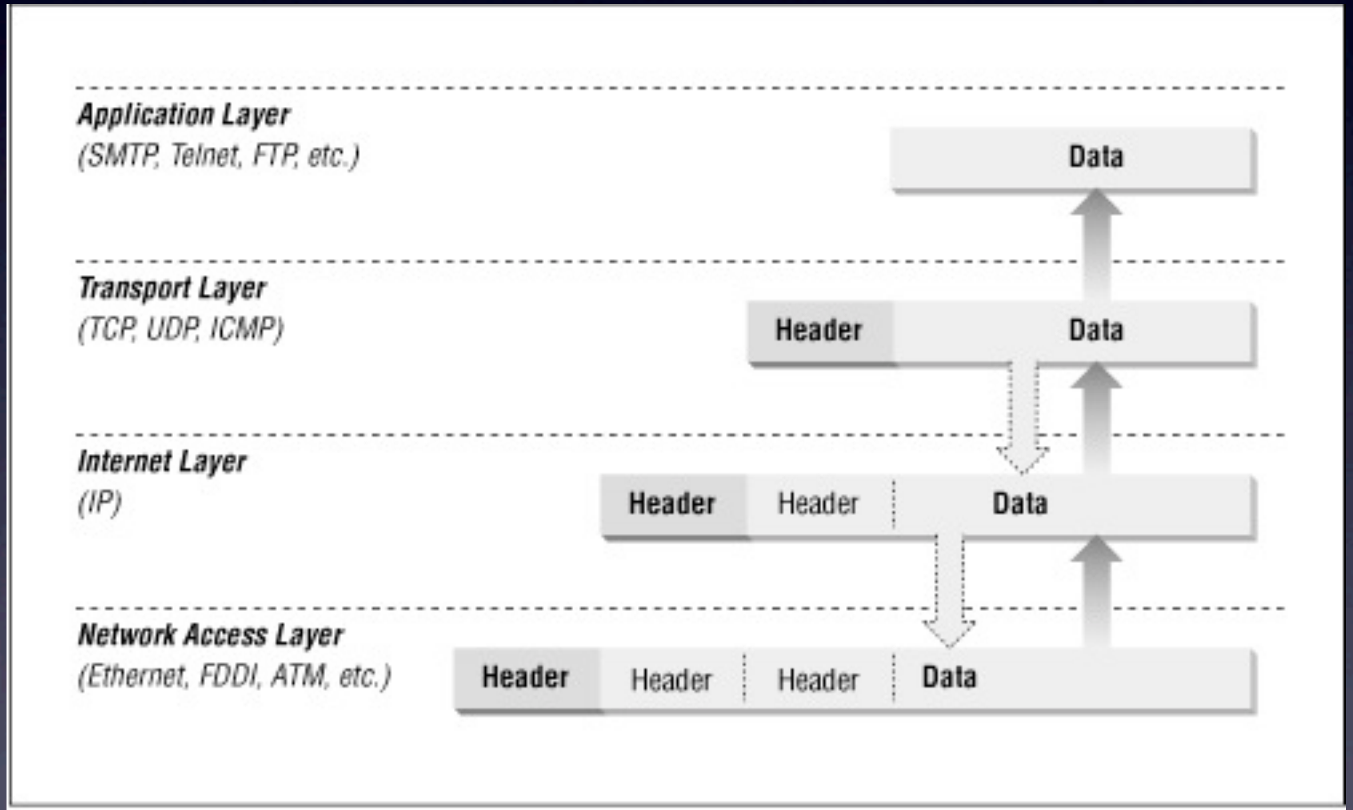
- Application/service-specific address for sending traffic to specific applications
- Port 80 is for HTTP
- Append them to an IP address like this: 24.49.212.123:80
- This specifies what computer, and application the traffic will go to

https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers

All together

- Computers connect using applications
- These applications use TCP and UDP protocols to transmit different types of data
- Both TCP and UDP sit on top of another layer called IP, which provides addressing and routing
- These connect via lower layers like the network access layer and physical access layer

- Don't worry about memorizing these



Servers

- Servers are systems that *respond* to requests across a computer network to provide access to a network service
- A single computer may provide access to several services on different ports

http://en.wikipedia.org/wiki/Server_(computing)

Client

- Clients initiate requests to a server to access a network service
- A computer can be a client *and* a server
- Your personal computer is often a client

https://en.wikipedia.org/wiki/Client_(computing)

Server Hardware

- Often dedicated and purpose-specific hardware
- Designed for long-term reliability, low power consumption and modularity
- Often located in a datacenter

Server Software

- Software that responds to requests
- Examples include Apache, Ngnix, Postgres and sshd
- Our software will run on top of these

HTTP

- Hypertext Transport Protocol
- Foundation of data communication for the World Wide Web
- Has several methods including: GET, POST, HEAD, PUT, DELETE, and PATCH
- This is the main protocol we'll interface with Ruby to make our server software
- Notice that this is what you type in your web browser bar!

http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol

HTTP Methods pt 1

- GET - Requests representation of specified source. Only retrieval and no other effect.
- HEAD - Same as GET, but without the body of the response
- POST - Adds a new resource. Non-indempotent
- PUT - Adds or updates (modifies) a resource. Indempotent.
- DELETE - Deletes specified resource

indempotent - multiple identical requests should have the same effect as a single request

HTTP Methods pt 2

- TRACE - Echos the request back. Used for testing for intermediary changes
- OPTIONS - Returns the HTTP methods that the server supports for specified URL.
- CONNECT - Converts the request connection to a transparent TCP/IP tunnel, usually to facilitate SSL-encrypted communication (HTTPS) through an unencrypted HTTP proxy.
- PATCH - Is used to apply partial modifications to a resource.

Note: These four are less frequently used.

HTTP Status Codes

- 200 - OK. Standard successful response
- 301 - Moved permanently. For redirecting moved content
- 401 - Unauthorized. Indicates improper authentication
- 404 - Not found. Resource could not be found.
- 500 - Generic server error

http://en.wikipedia.org/wiki/List_of_HTTP_status_codes

HTTP Headers

- Key-Value pairs (like hashes in Ruby!), but called fields
- They define the operating parameters of an HTTP transaction.
- Example keys: authorization (credentials), content-length, content-type (internet media type), cookie, date, referrer, user-agent
- This metadata is largely for the computer/browser.
  The body is intended to be rendered for the end-user.

http://en.wikipedia.org/wiki/Web_browser

Internet Media Types

- Also known as MIME types, or content types
- Identifier for file formats
- Helps browser figure out how to open/show the file
- Includes type and subtype
- Examples: audio/mpeg, text/html, image/jpeg, video/ogg

https://en.wikipedia.org/wiki/Internet_media_type

Web Browsers

- Make HTTP requests (GET, PUT, DELETE, etc)
- Render and interpret HTML, CSS and Javascript
- Provide user interface
- Examples: Chrome, Firefox, Safari, Internet Explorer

What is a Network?

What is an IP Address?

What is DNS?

What are the components of a URL?

What are the different HTTP methods/verbs?