Lecture 15
Networking Fundamentals

Slides attributed to
Neil Spring

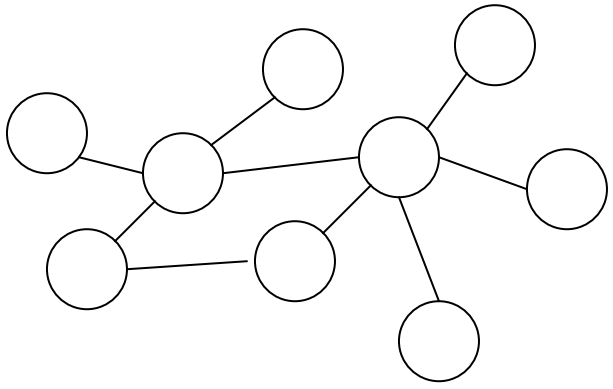
Today's Plan

- Talk about networking in general
  - Layers, Routing
- Specifically about IP and TCP
  - Service model, what TCP provides
- Work our way up to HTTP
  - What happens when you click a link?
- If time, drop down to Ethernet and wireless networking problems.
- Networks are distributed systems: failure is *expected*

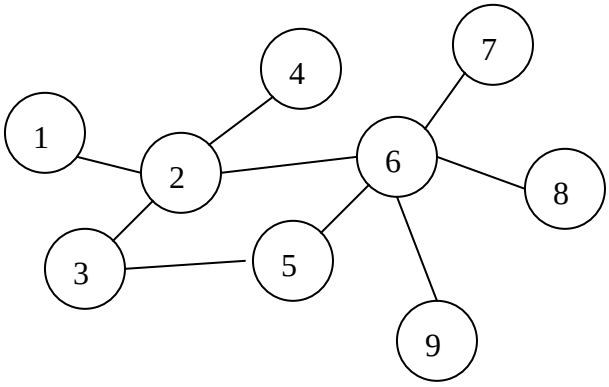
Basics

- Network – collection of nodes and links that cooperate for communication
- Nodes – computer systems
  - Internal (routers, bridges, switches)
  - Terminal (workstations)
- Links – connections for transmitting data
- Protocol – standards for formatting and interpreting data and control information

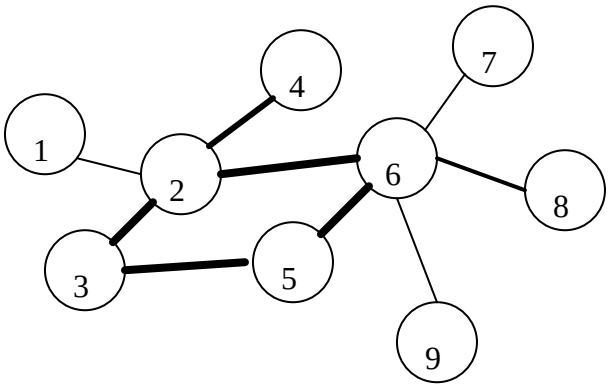
Network = {nodes and links}



Nodes have addresses



Links have bandwidths and latencies



Wires aren't perfect

- Attenuation (resistance)
  - degrades quality of signal
- Delay (speed of light * 2/3)
  - speed of light:
    - 8ms RTT coast-to-coast
    - 8 minutes to the sun
- Noise (microwaves and such)
- Nodes aren't perfect either
  - Unreliability is pervasive!

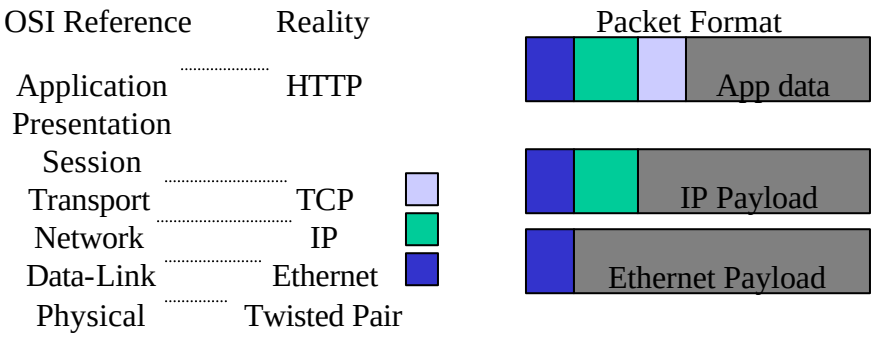
Getting Data Across (imperfect wires)

- Split up big files into small pieces
  - the pieces are called packets
- Each packet (~1500 bytes) is sent separately
  - packets can be corrupted
    - noise, bugs
  - packets can be dropped
    - corrupted, overloaded nodes
  - packets can be reordered
    - retransmission + different paths
- Allows packets from different flows to be multiplexed along the same link

Layers

- Each layer abstracts the services of various lower layers, providing a uniform interface to higher layers.
- Each layer needs to know:
  - How to interpret a packet's payload
    - e.g., protocol numbers
  - How to use the services of a lower layer
    - e.g., IP must know how to use ARP

Layers



The Internet Protocol (IP)

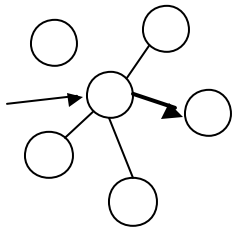
- Connects disparate networks
  - Single (hierarchical) address space
  - Single network header
- Assumes data link is unreliable
- Provides unreliable service
  - Loss: A B D E
  - Duplication: A B B C D E
  - Corruption: A Q C D E
  - Reordering: A C D B E

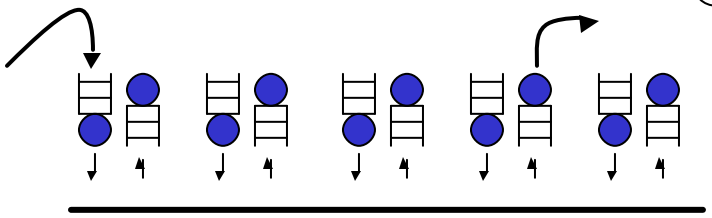
IP Addresses

- 32 bits long, split into 4 octets:
  - For example, 128.95.2.24
- Hierarchical:
  - First bits describe which network
  - Last bits describe which host on the network
- UW subnets include:
  - 128.95, 140.142…
- UW CSE subnets include:
  - 128.95.2, 128.95.4, 128.95.219…

Packet Forwarding

- Buffer incoming packets
- Decide which output link
- Buffer outgoing packets
- Send packet

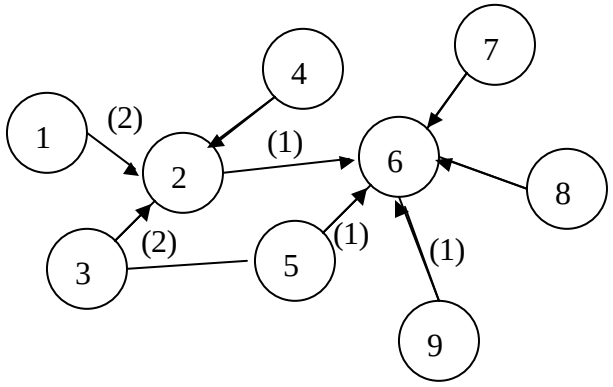




Routing

- How do nodes determine which output link to use to reach a destination?
- Distributed algorithm for converging on shortest path tree
- Nodes exchange reachability information:
  - "I can get to 128.95.2.x in 3 hops"

Shortest path tree



(x) Is the cost to get to 6. The metric (cost per link) here is 1. Simple algorithm: 6 broadcasts "I'm alive" to neighbors. Neighbors send "I can get to 6 in 1 hop", etc.

Route Aggregation

- What hierarchical addressing is good for.
- UW routers can advertise 128.95.x.x
  - instead of 128.95.2.x, 128.95.3,x, …
- Other routers don't need forwarding table entries for each host in the network.

Routing Reality

- Routing in the Internet connects Autonomous Systems (AS's)
  - AT&T, Sprint, UUNet, BBN…
- Shortest path, sort of… money talks.
  - actually a horrible mess
  - nobody really knows what's going on
  - get high $$ job if you are a network engineer that messes with this stuff

TCP Service Model

- Provide reliability, ordering on the unreliable, unordered IP
- Bytestream oriented: when you send data using TCP, you think about bytes, not about packets.

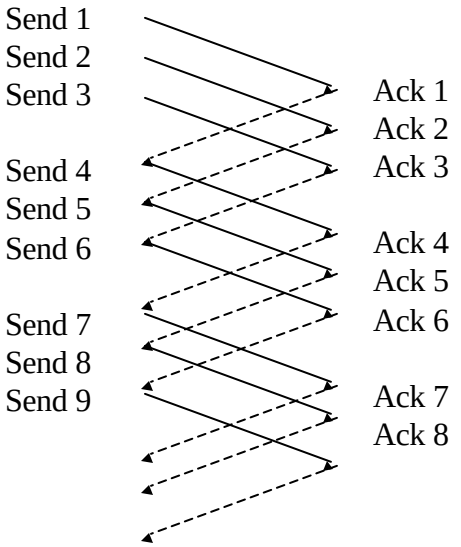
TCP Ports

- Connections are identified by the tuple:
  - IP source address
  - IP destination address
  - TCP source port
  - TCP destination port
- Allows multiple connections; multiple application protocols, between the same machines
- Well known ports for some applications: (web: 80, telnet: 23, mail:25, dns: 53)

TCP's Sliding Window

- Simple reliability:
  - Send one packet, wait for acknowledgment, then send the next…
- Better performance:
  - Keep several unacknowledged packets in the network (a window)

Sliding Window Example



- Window size = 3
- Can send up to three packets into the network at a time.
- Each packet has a sequence number for ordering

TCP's Congestion Control

- How big should the window be?
- Performance is limited by:
  - (window size) / round trip time
  - Performance of bottleneck link (modem?)
- If window is too small, performance is wasted.
- If window is too big, may overflow network buffers, causing packet loss.

Steps for a web access

- Name lookup
  - Client to local DNS server
  - Local DNS may return a cached binding, or lookup the name for itself
- TCP Connection setup
  - Client to remote IP, port 80
- Send HTTP request
  - "GET /index.html"
- Receive HTTP response
  - "blah blah blah" maybe several packets
- TCP Connection teardown

HTTP 1.1

- Incremental improvements
- "Persistent connections" allow multiple requests over the same connection
  - Web transfers are often small
  - Avoid connection setup and teardown overhead
  - TCP is better the longer you use it: it learns how fast to send to get best performance without overflowing buffers.

Ethernet

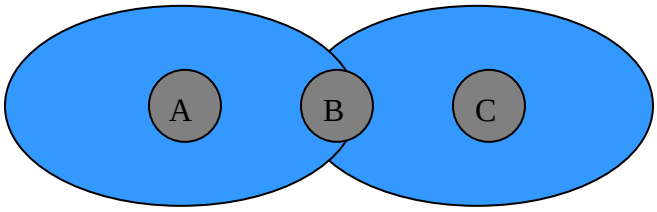
- CSMA/CD
- Carrier Sense: everyone listens until silence before speaking
- Multiple Access: everyone talks on the same wire
- Collision Detection: listen while talking to abort when two talk at the same time

Wireless networking

- Can we apply CSMA/CD to wireless networks too?
- What's different about networks without wires
  - More attenuation (fading)
  - More noise (burst errors too)
  - More delay (possibly)
  - Interference (walls, mountain ranges…)

802.11 Wireless

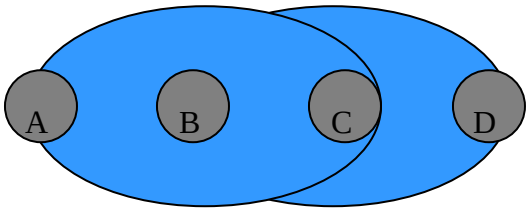
- Hidden station problem



- A sends to B, C might interrupt!

802.11 Wireless II

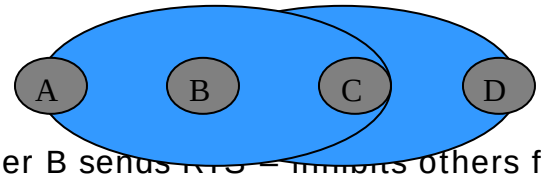
- Exposed station problem



- B could send to A and C to D at the same time (but would be inhibited if detecting collision)

802.11 Collision Avoidance

- Multiple Access Collision Avoidance



- Sender B sends RTS – inhibits others from sending for a short time
- Receiver C sends CTS – inhibits others from sending until packet transferred.
- Sender B sends the packet.