Links

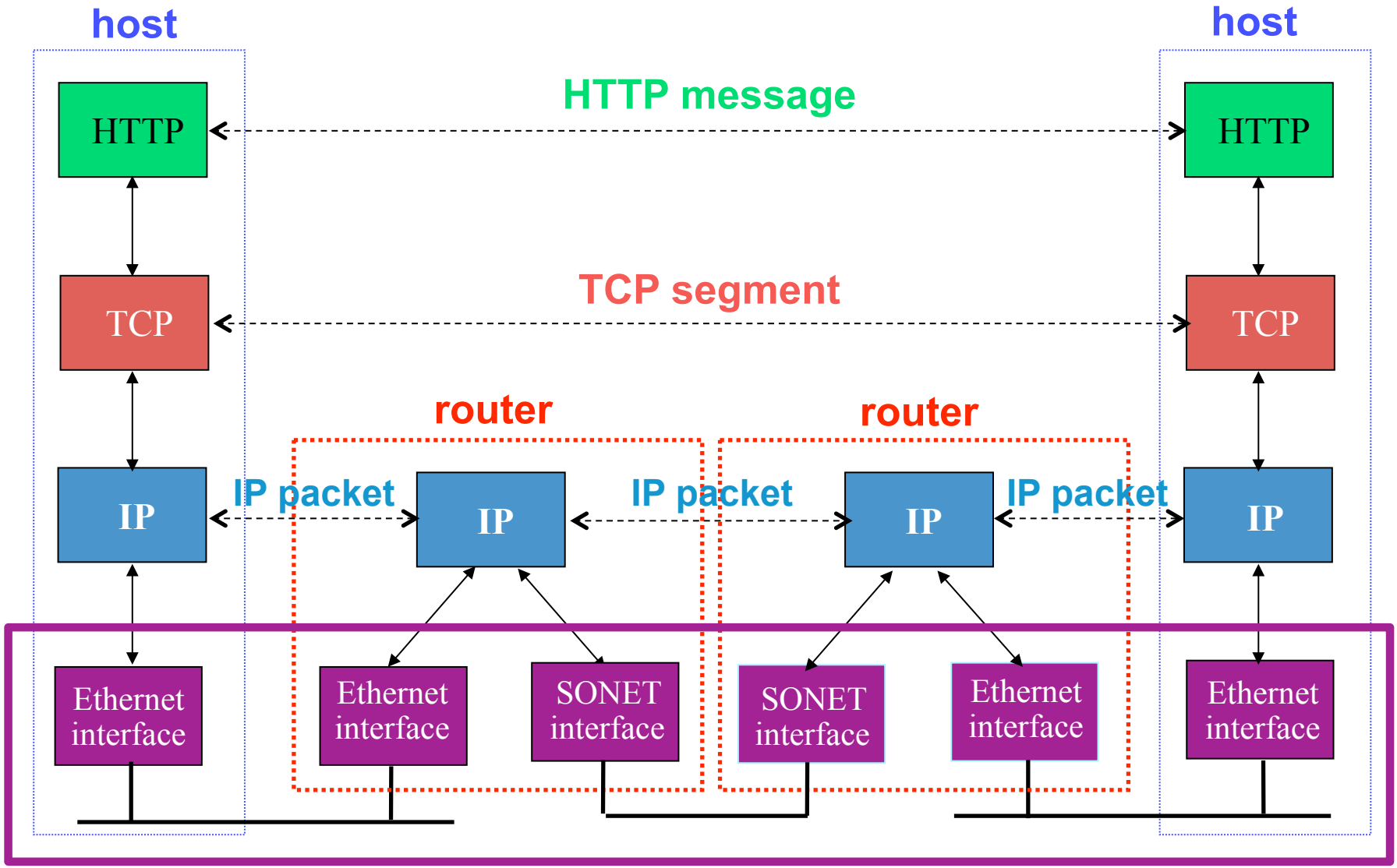
COS 461: Computer Networks
Spring 2011

Mike Freedman
http://www.cs.princeton.edu/courses/archive/spring11/cos461/

Outline

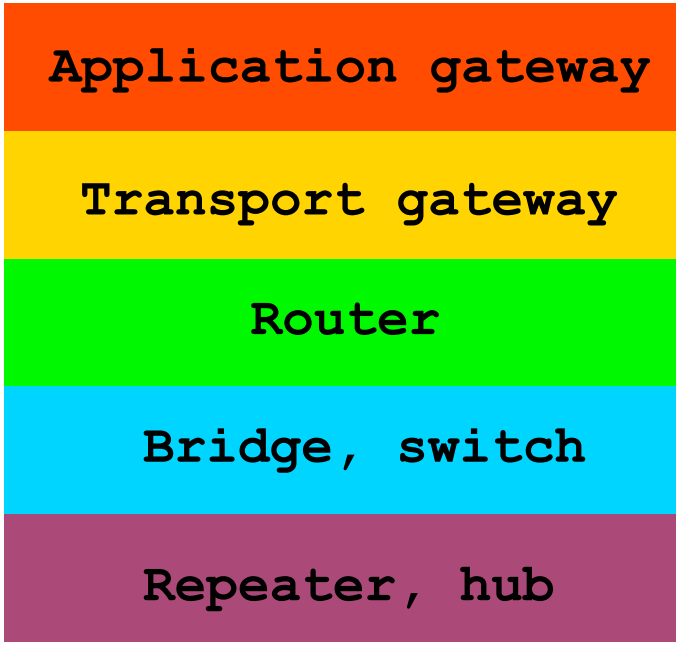
- Link-layer protocols
  - Encoding, framing, error detection
- Multiple-access links: sharing is caring!
  - Strict isolation: division over time or frequency
  - Centralized management (e.g., token passing)
  - Decentralized management (e.g., CSMA/CD)
- 2nd precept this Friday
  - Everybody assigned last week; attendance mandatory
  - Assignment 0 due Friday in precept

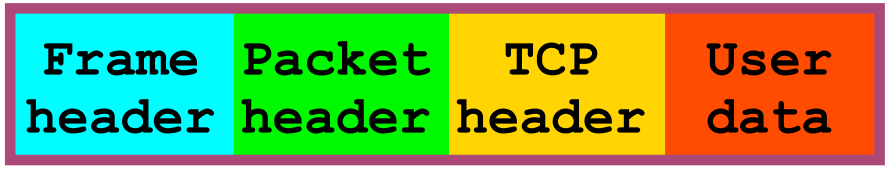
Recall the Internet layering model



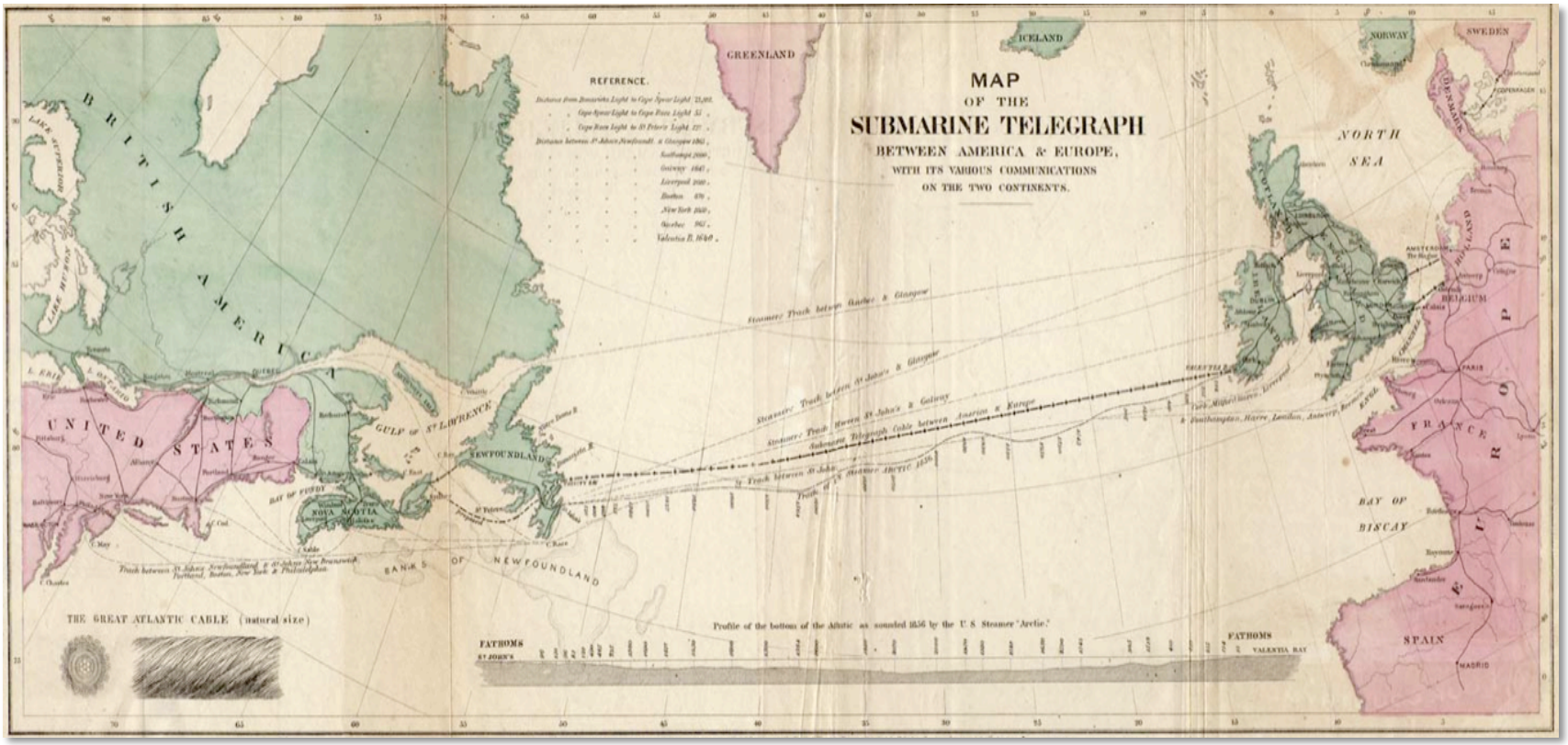
Shuttling Data at Different Layers

- Different devices switch different things
  - Network layer: packets (routers)
  - Link layer: frames (bridges and switches)
  - Physical layer: electrical signals (repeaters and hubs)





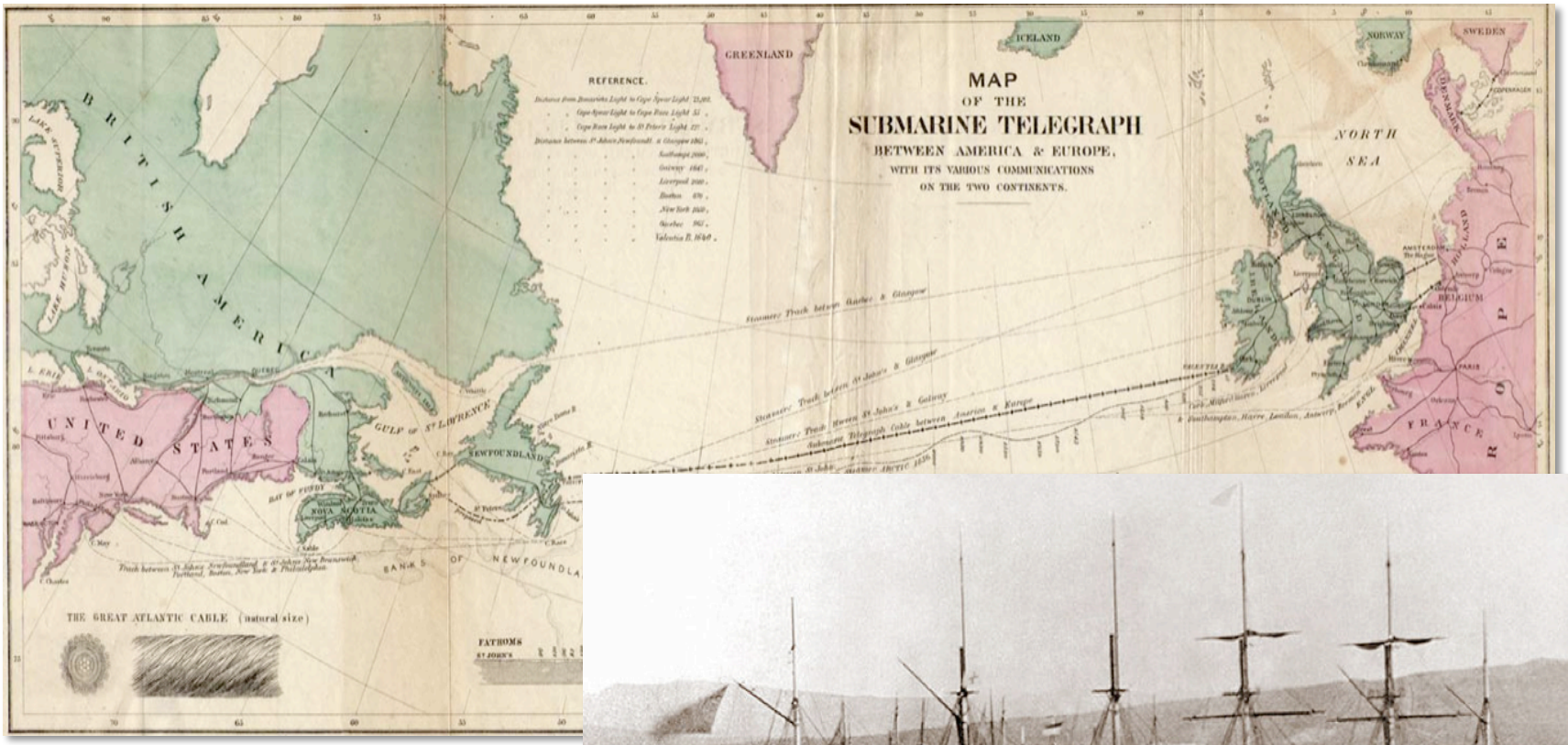
Link Transmissions in 1858



On August 16, 1858: "Glory to God in the highest; on earth, peace and good will toward men."

17 hours to transmit, 2 min per character, 0.1 words per minute

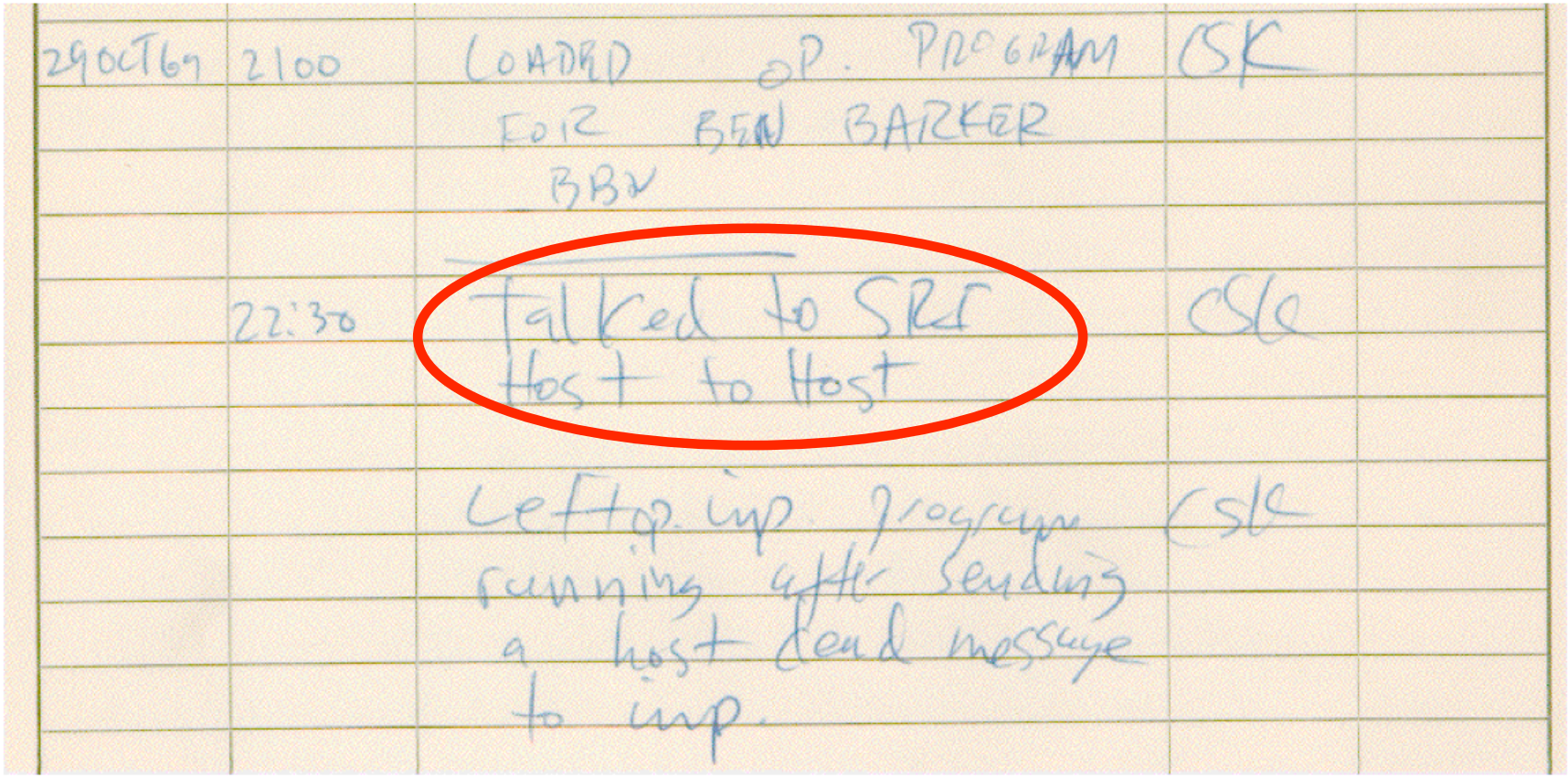
Link Transmissions in 1866



1866 cable:
8 words / minute

Great Eastern – cable laying ship

IP Transmissions in 1969



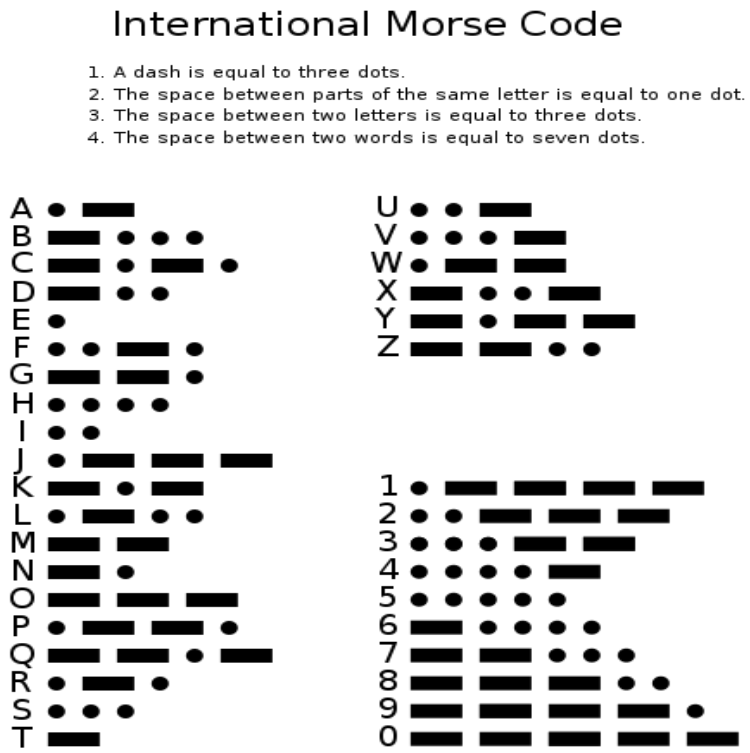
First message on Internet:

$ lo >system crash<

Digital -> Analog Encoding

- Signals sent over physical links
  - Source node: bits -> signal
  - Receiving node: signal -> bits

- Encoding in telegraph
  - Morse code: “long” and “short” signals



Digital -> Analog Encoding

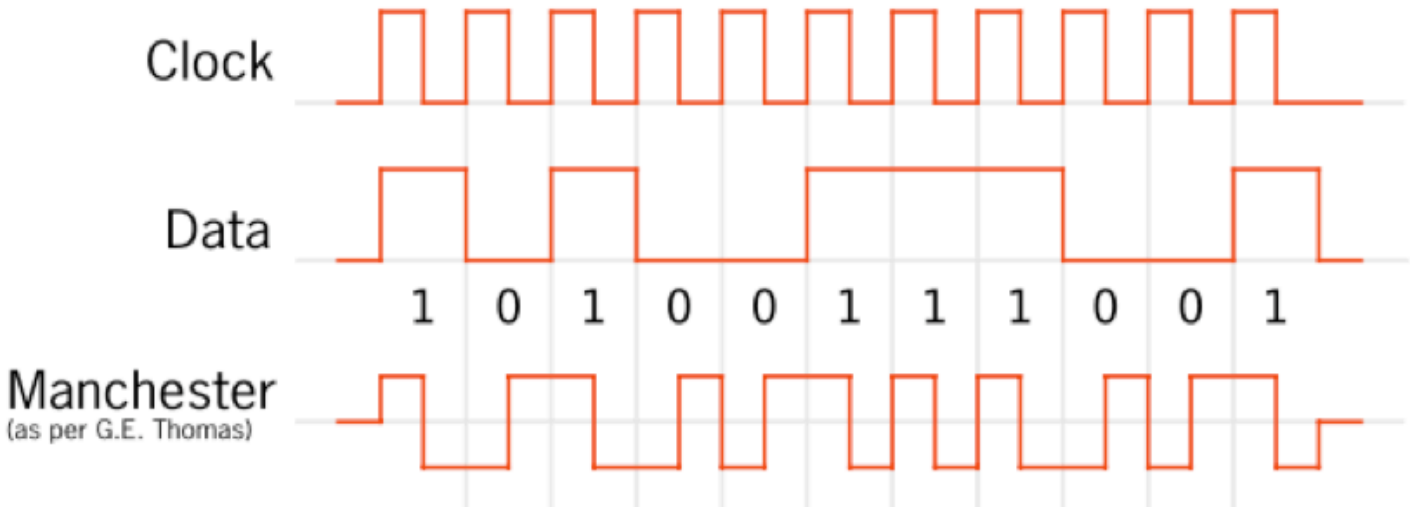
- Signals sent over physical links
  - Source node: bits -> signal
  - Receiving node: signal -> bits
- Simplify some electrical engineering details
  - Assume two discrete signals, high and low
  - E.g., could correspond to two different voltages
- Simple approach: Non-return to zero
  - High for a 1, low for a 0

Problem With NRZ

- Long strings of 0s or 1s introduce problems
  - No transitions from low-to-high, or high-to-low
- Receiver keeps average of signal it has received
  - Uses the average to distinguish between high and low
  - Long flat strings make receiver sensitive to small change
- Transitions also necessary for clock recovery
  - Receiver uses transitions to derive its own clock
  - Long flat strings do not produce any transitions
  - Can lead to clock drift at the receiver

Protocols with clock-recovery

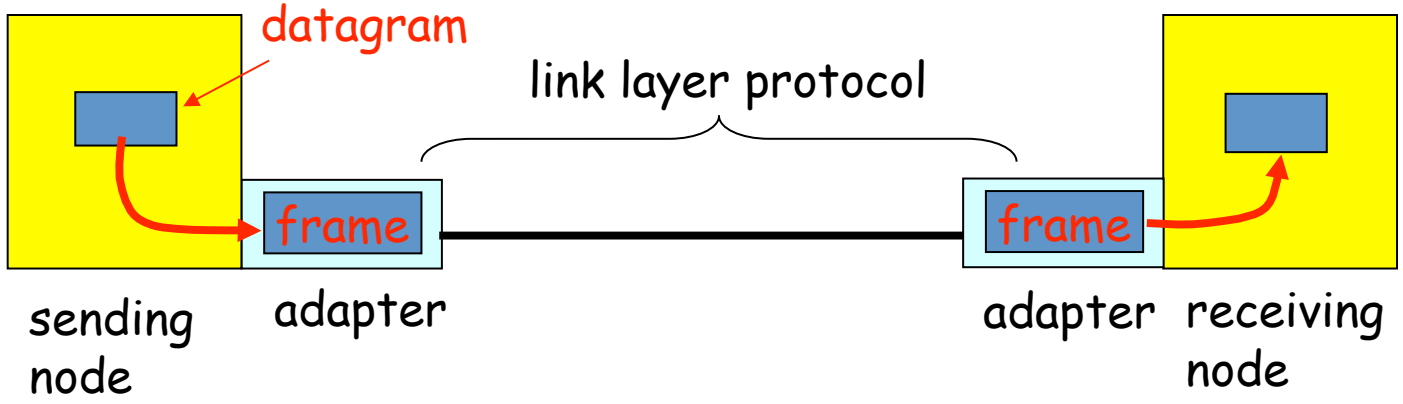
- Manchester encoding (basic Ethernet)
  - clock XOR NRZ: L→H (0), H→L (1) : self-clocking



- Efficiency?
  - Manchester: 2 clock transitions per bit: 50% efficient
  - 1 GbE: 8b/10b: 80% efficient
  - 10 GbE: 64b/66b: 96.9% efficient

Link-Layer Protocols

Digital adaptors Communicating



- Link layer implemented in adaptor (network interface card)
  - Ethernet card, PCMCIA card, 802.11 card
- Sending side:
  - Encapsulates datagram in a frame
  - Adds error checking bits, flow control, etc.
- Receiving side
  - Looks for errors, flow control, etc.
  - Extracts datagram and passes to receiving node

Link-Layer Services

- Encoding
  - Representing the 0s and 1s
- Framing
  - Encapsulating packet into frame, adding header, trailer
  - Using MAC addresses, rather than IP addresses
- Error detection
  - Errors caused by signal attenuation, noise.
  - Receiver detecting presence of errors
- Error correction
  - Receiver correcting errors without retransmission
- Flow control
  - Pacing between adjacent sending and receiving nodes

Framing

- Break sequence of bits into a frame
  - Typically implemented by the network adaptor
- Sentinel-based
  - Delineate frame with special pattern (e.g., 01111110)

  - Problem: what if special patterns occurs within frame?
  - Solution: escaping the special characters
    - E.g., sender always inserts a 0 after five 1s
    - … and receiver always removes a 0 appearing after five 1s
  - Similar to escaping special characters in C programs

Framing (Continued)

- Counter-based
  - Include the payload length in the header
  - … instead of putting a sentinel at the end
  - Problem: what if the count field gets corrupted?
    - Causes receiver to think the frame ends at a different place
  - Solution: catch later when doing error detection
    - And wait for the next sentinel for the start of a new frame
- Clock-based
  - Make each frame a fixed size
  - No ambiguity about start and end of frame
  - But, may be wasteful

Error Detection

- Errors are unavoidable
  - Electrical interference, thermal noise, etc.
- Error detection
  - Transmit extra (redundant) information
  - Use redundant information to detect errors
  - Extreme case: send two copies of the data
  - Trade-off: accuracy vs. overhead

Error Detection Techniques

- Parity check
  - Add an extra bit to a 7-bit code
  - Odd parity: ensure an odd number of 1s
    - E.g., 0101011 becomes 01010111
  - Even parity: ensure an even number of 1s
    - E.g., 0101011 becomes 01010110
- Checksum
  - Treat data as a sequence of 16-bit words
  - Compute a sum of all 16-bit words, with no carries
  - Transmit the sum along with the packet
- Cyclic Redundancy Check (CRC)
  - See Section 2.4.3

Towards Concurrency

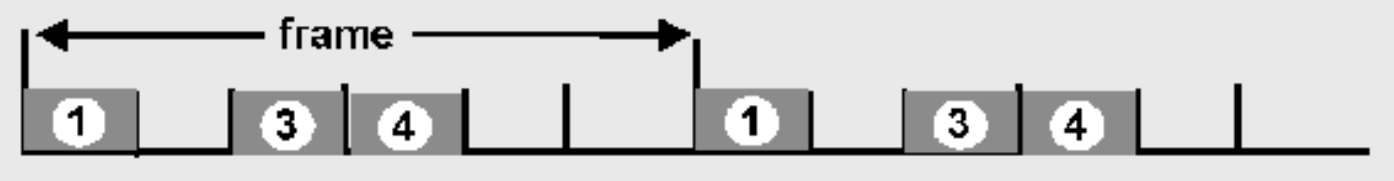
Multiple Access Protocol

- Single shared broadcast channel
  - Avoid having multiple nodes speaking at once
  - Otherwise, collisions lead to garbled data
- Multiple access protocol
  - Distributed algorithm for sharing the channel
  - Algorithm determines which node can transmit
- Classes of techniques
  - Channel partitioning: divide channel into pieces
  - Taking turns: passing a token for the right to transmit
  - Random access: allow collisions, and then recover

Channel Partitioning: TDMA

TDMA: time division multiple access

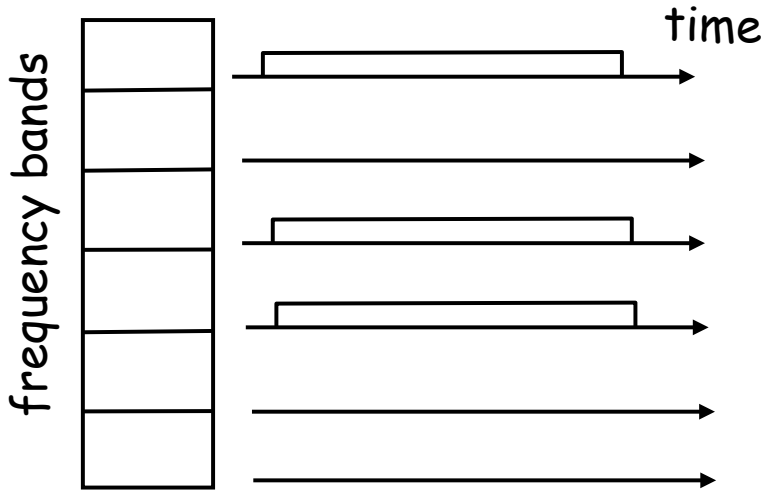
- Access to channel in "rounds"
  - Each station gets fixed length slot in each round
- Time-slot length is packet transmission time
  - Unused slots go idle
- Example: 6-station LAN with slots 1, 3, and 4



Channel Partitioning: FDMA

FDMA: frequency division multiple access

- Channel spectrum divided into frequency bands
  - Each station has fixed frequency band (Wifi channels 1-11)
- Unused transmission time in bands go idle
- Example: 6-station LAN with bands 1, 3, and 4



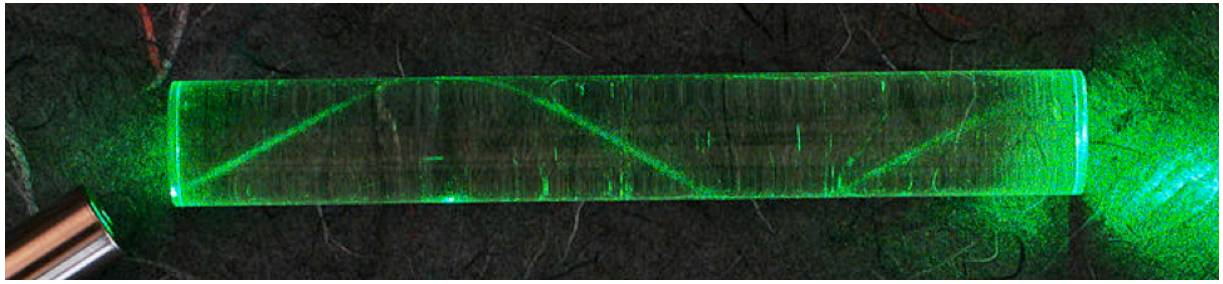
Channel Partitioning: FDMA

FDMA: frequency division multiple access

- Channel spectrum divided into frequency bands
  - Each station has fixed frequency band (Wifi channels 1-11)
- Unused transmission time in bands go idle
- Example: 6-station LAN with bands 1, 3, and 4

WDM: Wavelength division multiplexing

- Multiple wavelengths λ on same optical fiber



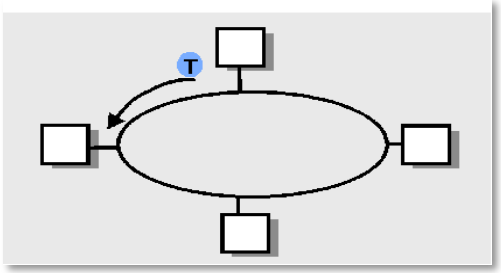
"Taking Turns" MAC protocols

Polling

- Primary node "invites" secondary nodes to transmit in turn
- Concerns:
  - Polling overhead
  - Latency
  - Single point of failure (primary)

Token passing

- Control token passed from one node to next sequentially
- Token message
- Concerns:
  - Token overhead
  - Latency
  - Single point of failure (token)



Random Access Protocols

- When node has packet to send
  - Transmit at full channel data rate R.
  - No *a priori* coordination among nodes
- Two or more transmitting nodes ➜ "collision"
- Random access MAC protocol specifies:
  - How to detect collisions
  - How to recover from collisions

Key Ideas of Random Access

- Carrier Sense (CS)
  - *Listen before speaking, and don't interrupt*
  - Checking if someone else is already sending data
  - … and waiting till the other node is done
- Collision Detection (CD)
  - *If someone else starts talking at the same time, stop*
  - Realizing when two nodes are transmitting at once
  - …by detecting that the data on the wire is garbled
- Randomness
  - *Don't start talking again right away*
  - Waiting for a random time before trying again

Slotted ALOHA

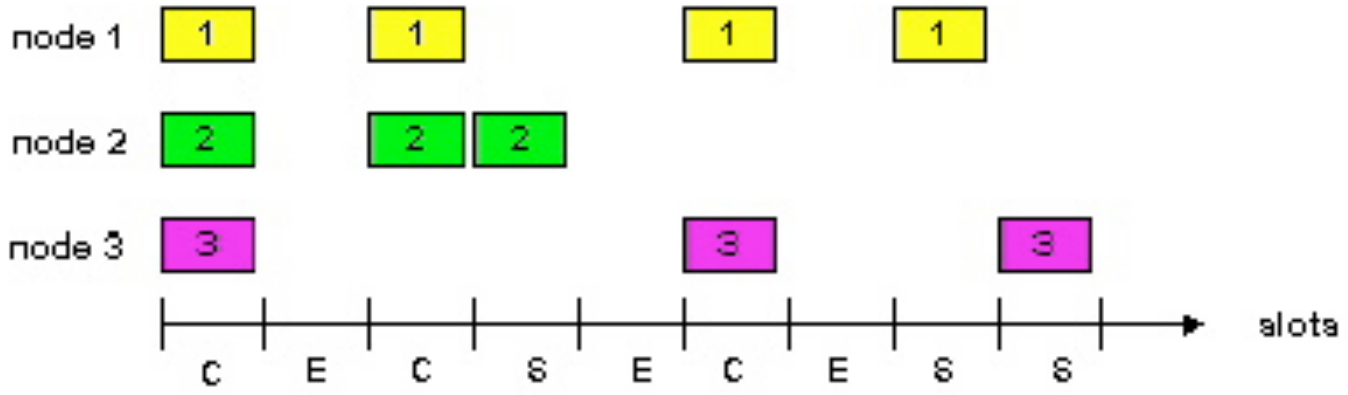
Assumptions

- All frames same size
- Time divided into equal slots (time to transmit a frame)
- Nodes start to transmit frames only at start of slots
- Nodes are synchronized
- If two or more nodes transmit, all nodes detect collision

Operation

- When node obtains fresh frame, transmits in next slot
- No collision: node can send new frame in next slot
- Collision: node retransmits frame in each subsequent slot with probability *p* until success

Slotted ALOHA



Pros

- Single active node can continuously transmit at full rate of channel
- Highly decentralized: only slots in nodes need to be in sync
- Simple

Cons

- Collisions, wasting slots
- Idle slots
- Nodes may be able to detect collision in less than time to transmit packet
- Clock synchronization

CSMA (Carrier Sense Multiple Access)

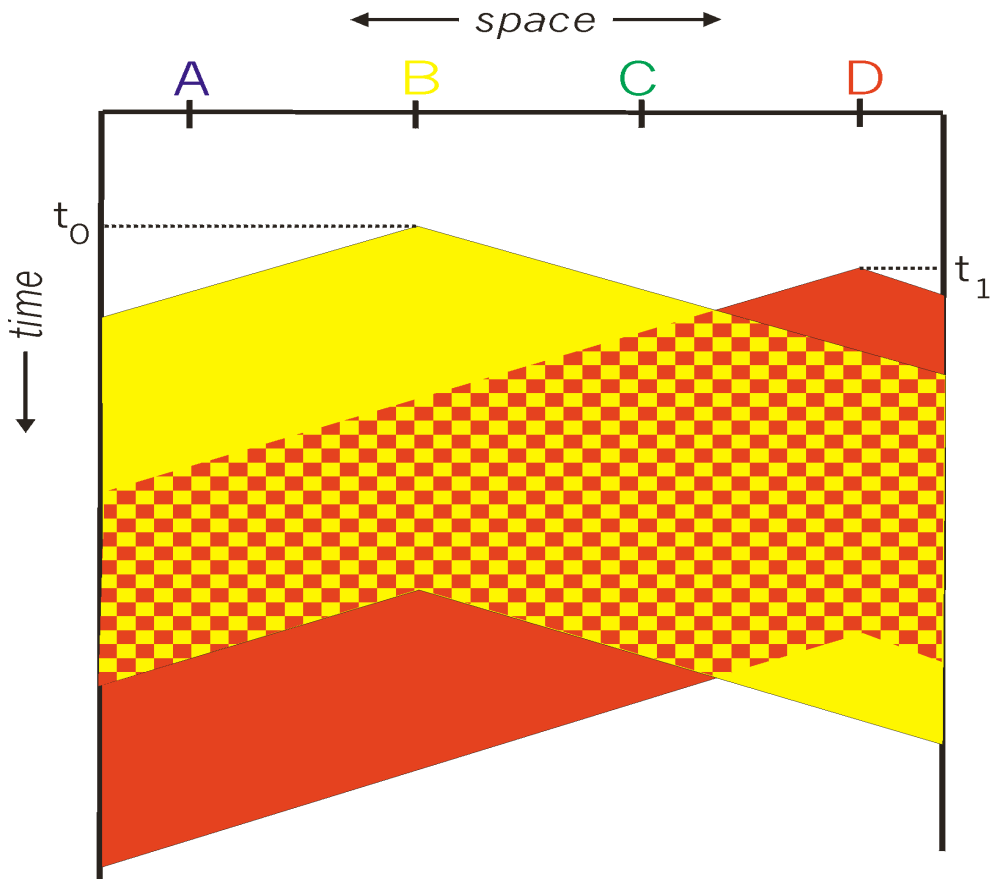
- Collisions hurt the efficiency of ALOHA protocol
  - At best, channel is useful 37% of the time
- CSMA: listen before transmit
  - If channel sensed idle: transmit entire frame
  - If channel sensed busy, defer transmission
- Human analogy: don't interrupt others!

CSMA (Carrier Sense Multiple Access)

CSMA: Listen before transmit

Collisions *can* still occur: propagation delay means two nodes may not hear each other's transmission

Collision: entire packet transmission time wasted

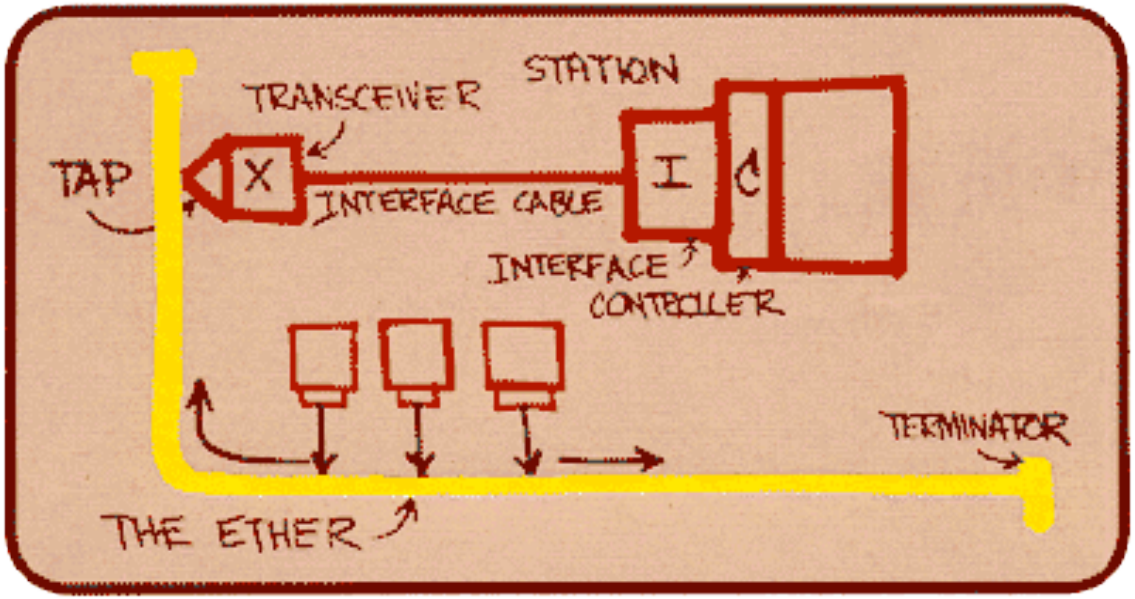


CSMA/CD (Collision Detection)

- CSMA/CD: carrier sensing, deferral as in CSMA
  - Collisions detected within short time
  - Colliding transmissions aborted, reducing wastage
- Collision detection
  - Easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - Difficult in wireless LANs: receiver shut off while transmitting
- Human analogy: the polite conversationalist

Ethernet

- Dominant wired LAN technology
- First widely used LAN technology
- Simpler, cheaper than token LANs and ATM
- Kept up with speed race: 10 Mbps – 10 Gbps

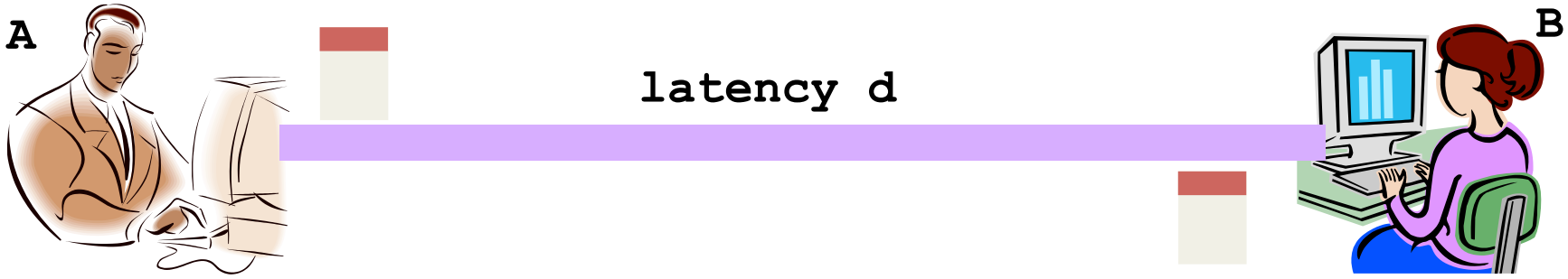


Metcalfe's Ethernet sketch

Ethernet Uses CSMA/CD

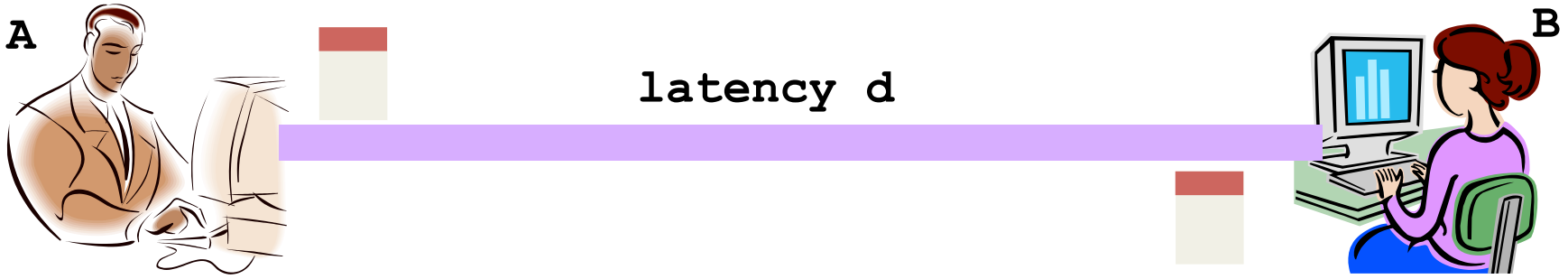
- Carrier Sense: wait for link to be idle
  - Channel idle: start transmitting
  - Channel busy: wait until idle
- Collision Detection: listen while transmitting
  - No collision: transmission is complete
  - Collision: abort transmission, and send jam signal
- Random access: exponential back-off
  - After collision, wait a random time before trying again
  - After m^{th} collision, choose K randomly from {0, …, 2^m-1}
  - … and wait for K*512 bit times before trying again

Limitations on Ethernet Length



- Latency depends on physical length of link
  - Time to propagate a packet from one end to the other
- Suppose A sends a packet at time t
  - And B sees an idle line at a time just before t+d
  - … so B happily starts transmitting a packet
- B detects a collision, and sends jamming signal
  - But A doesn't see collision till t+2d

Limitations on Ethernet Length



- A needs to wait for time 2d to detect collision
  - So, A should keep transmitting during this period
  - … and keep an eye out for a possible collision
- Imposes restrictions on Ethernet
  - Maximum length of the wire: 2500 meters
  - Minimum length of the packet: 512 bits (64 bytes)

Three Ways to Share the Media

- Channel partitioning MAC protocols:
  - Share channel efficiently and fairly at high load
  - Inefficient at low load: delay in channel access, 1/N bandwidth allocated even if only 1 active node!
- "Taking turns" protocols
  - Eliminates empty slots without causing collisions
  - Vulnerable to failures (e.g., failed node or lost token)
- Random access MAC protocols
  - Efficient at low load: single node can fully utilize channel
  - High load: collision overhead