CS4450

Computer Networks:
Architecture and Protocols

Lecture 6
Data Link Layer

Prof. Rachit Agarwal

Announcements

- Problem Set 1 solutions released (Ed Discussions)
- Problem Set 2 released (Course website)

Context for Today's Lecture

- You now understand
  - Network sharing (in depth)
  - Architectural principles (in depth)
  - Design goals for the Internet (& computer networks, in depth)
  - End-to-end working of the Internet (at a high-level)

- Now, time to dive deeper:
  - Link Layer (~1 week)
  - Network Layer (~4 weeks)
  - Transport Layer (~3 weeks)

- **Today: Datalink (or, simply Link) Layer**

Goals for Today's Lecture

- **Link layer:**
  - **Broadcast medium**
  - Sharing broadcast medium
  - Carrier Sense Multiple Access - Collision Detection (CSMA/CD)

Data Link Layer

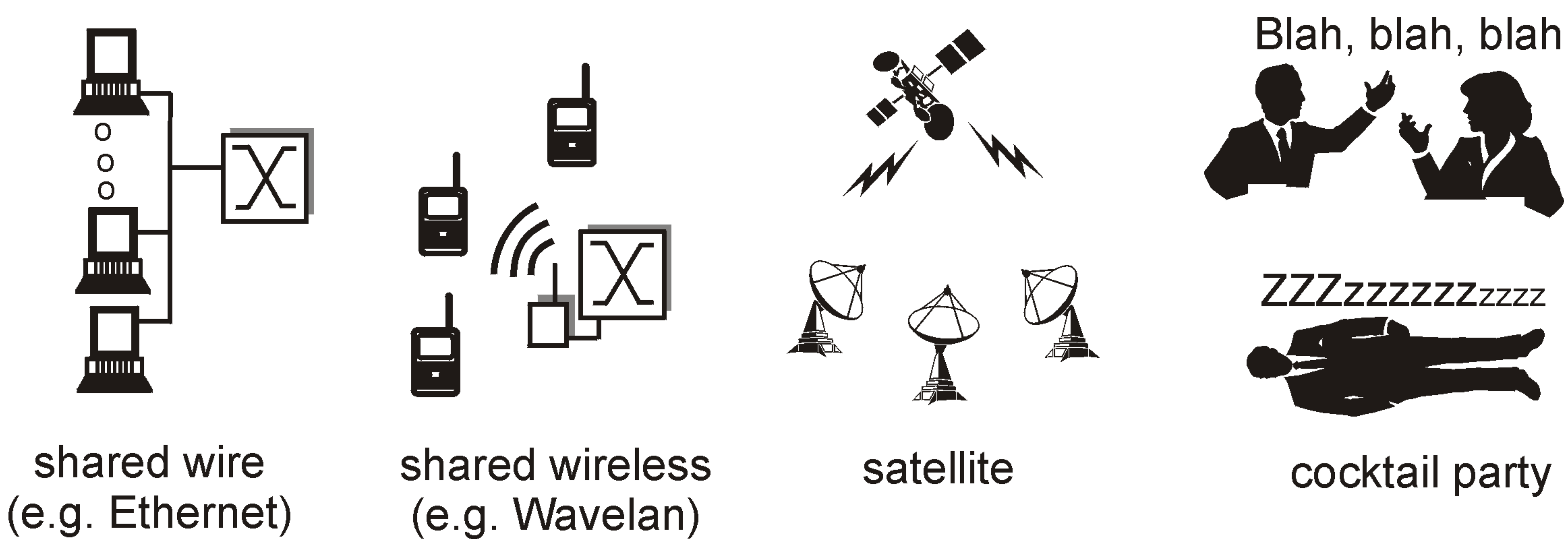
Data Link Layer

- **Two types of communication mediums**
  - **Point-to-point**
    - The high-level ideas discussed so far were for point-to-point
  - **Broadcast**
    - Original design of Link layer protocols
    - More recent versions have moved to point-to-point
      - We will discuss why so!

- **Network Adapters (e.g., NIC — network interface card)**
  - The hardware that connects a machine to the network
  - Has a “name” — MAC (Medium access control) address

Point-to-Point vs. Broadcast Medium

- Point-to-point: **dedicated** pairwise communication
  - E.g., long distance fiber link
  - E.g., Point-to-point link between two routers
- Broadcast: **shared** wire or medium
  - Traditional Link Layer (Ethernet)
  - 802.11 wireless LAN

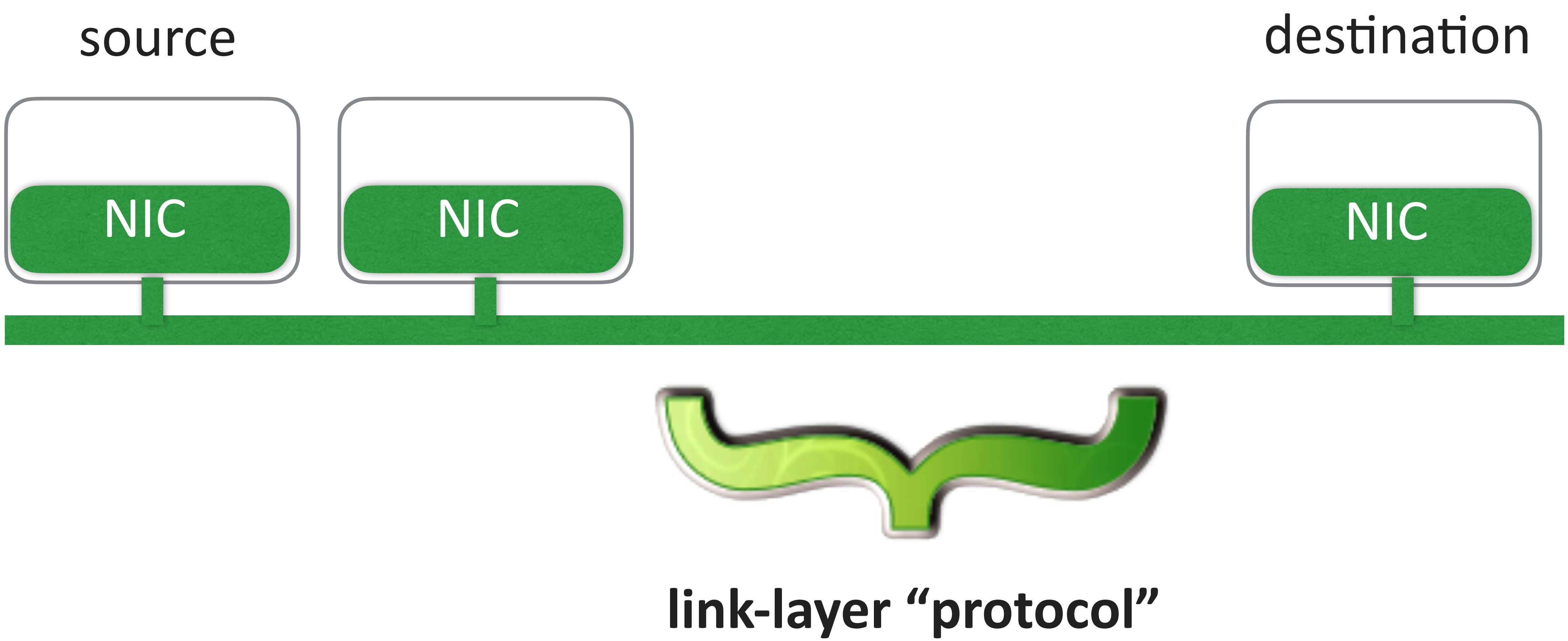


Data Link Layer: Broadcast (until ~2000s)

- **Ever been to a party?**
  - Tried to have an interesting discussion?
- **Fundamental challenge?**
  - **Collisions**

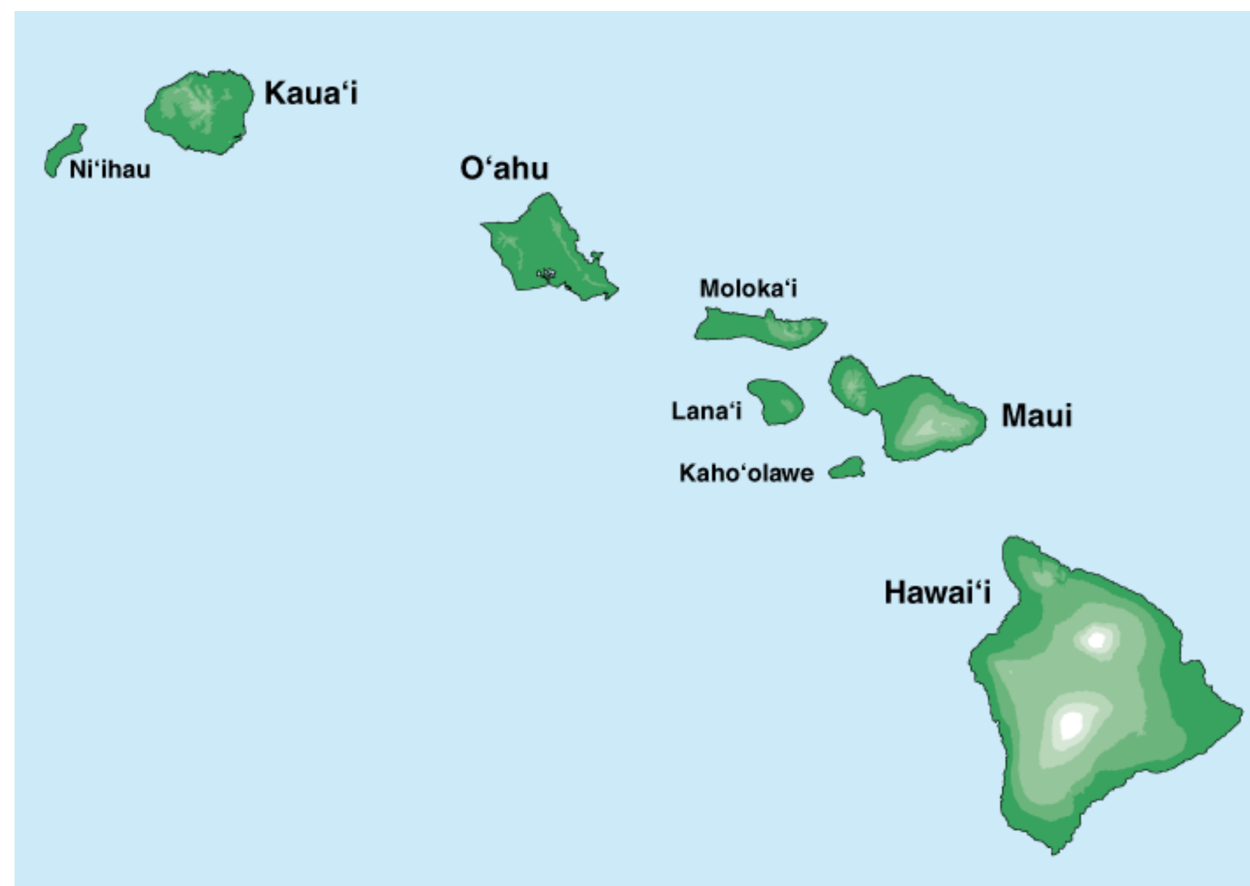
Broadcast Medium: Desirable properties

- **One and only one: data delivery**
- How do we design a broadcast medium protocol for data delivery?



Where it all Started: AlohaNet

- **Norm Abramson:**
  - Left Stanford in 1970
    - So he could SURF
  - Set up first data communication system for Hawaiian islands
  - Central hub at University of Hawaii, Oahu



Aloha Signaling

- Two channels: random access, broadcast

- Sites send packets to hub
  - Random access channel
  - Each site transmits packets at “random” times
  - If a packet not received (due to collision), site resends

- Hub sends packets to all sites
  - Broadcast channel
  - Sites can receive even if they are also sending

- **Challenge: Requires a centralized hub**
  - If the hub fails, the entire network fails
  - Not always a good design (remember the design goals?)

Sharing a broadcast channel

- **Context: a shared broadcast channel**
  - Must avoid/handle having multiple sources speaking at once
  - Otherwise collisions lead to garbled data
  - Need **distributed algorithm** for sharing channel
  - Algorithm determines **when** and **which** source can transmit

- **Three classes of techniques**
  - **Frequency-division multiple access**: divide channel into pieces
  - **Time-division multiple access**: divide channel into time slots
  - **Random access**: allow uncoordinated access
    - Detect collisions, and if needed, recover from collisions
    - More in the Internet style!

Frequency-Division Multiple Access (FDMA)

- **Frequency sharing**
  - Divide the channel into **frequencies**
  - **Every source is assigned a subset of frequencies**
    - And transmits data only on its assigned frequency
- **Goods: no collisions**
- **Not-so-good:**
  - A source may have nothing to send (frequency wasted)
  - Interference may cause disruption
  - Hard to implement for wired networks
- Used in many wireless networks
  - E.g., radio

Time-Division Multiple Access (TDMA)

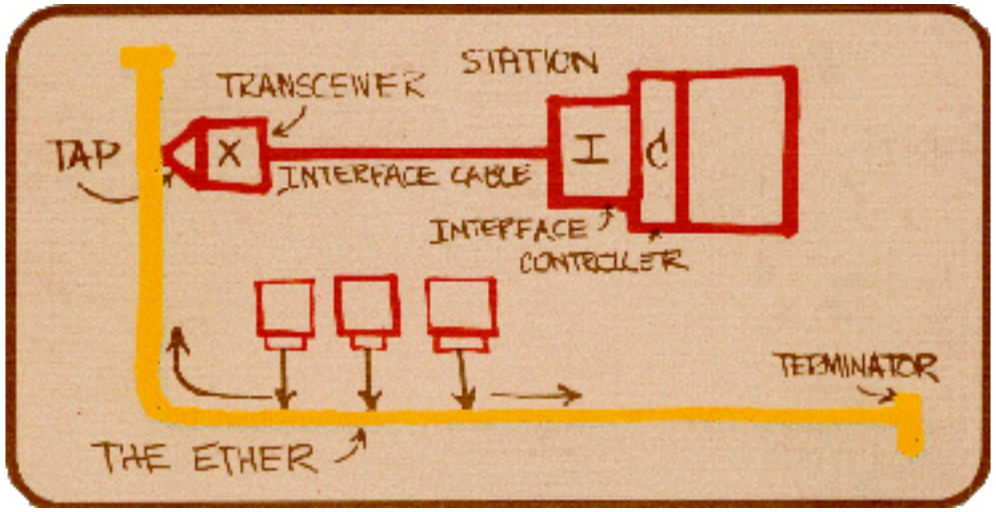
- **Time sharing**
  - Divide time into **slots**
  - Divide data into **frames**
    - Such that a frame can be transmitted in one slot
  - **Every source is assigned a subset of slots**
    - And transmits a frame only in its assigned slot

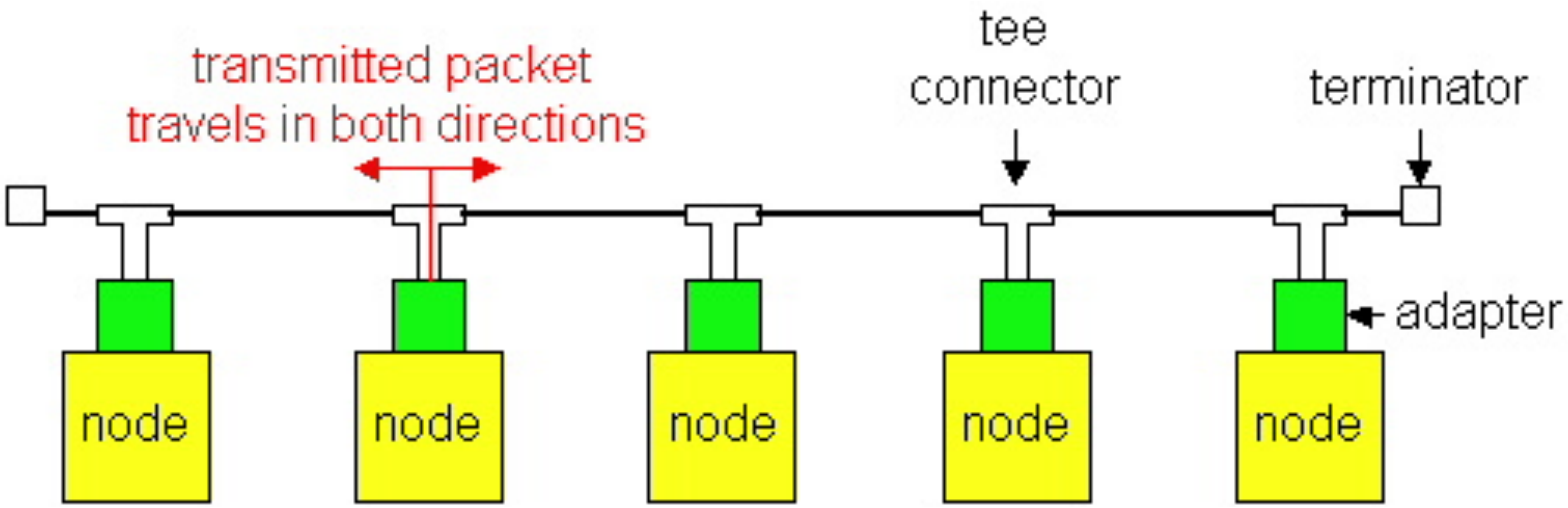
- **Goods: no collisions**

- **Not-so-good: Underutilization of resources**
  - During a slot, a source may have nothing to send
  - When the source has something to send, wait for its slot

Random Access

- **Bob Metcalfe:**
  - Xerox PARC
  - Visits Hawaii, and gets the idea
  - Shared wired medium





Life lesson:

If you want to invent great things,
go to Hawaii :-)

Link Layer (Media Access Control, or MAC) Protocol

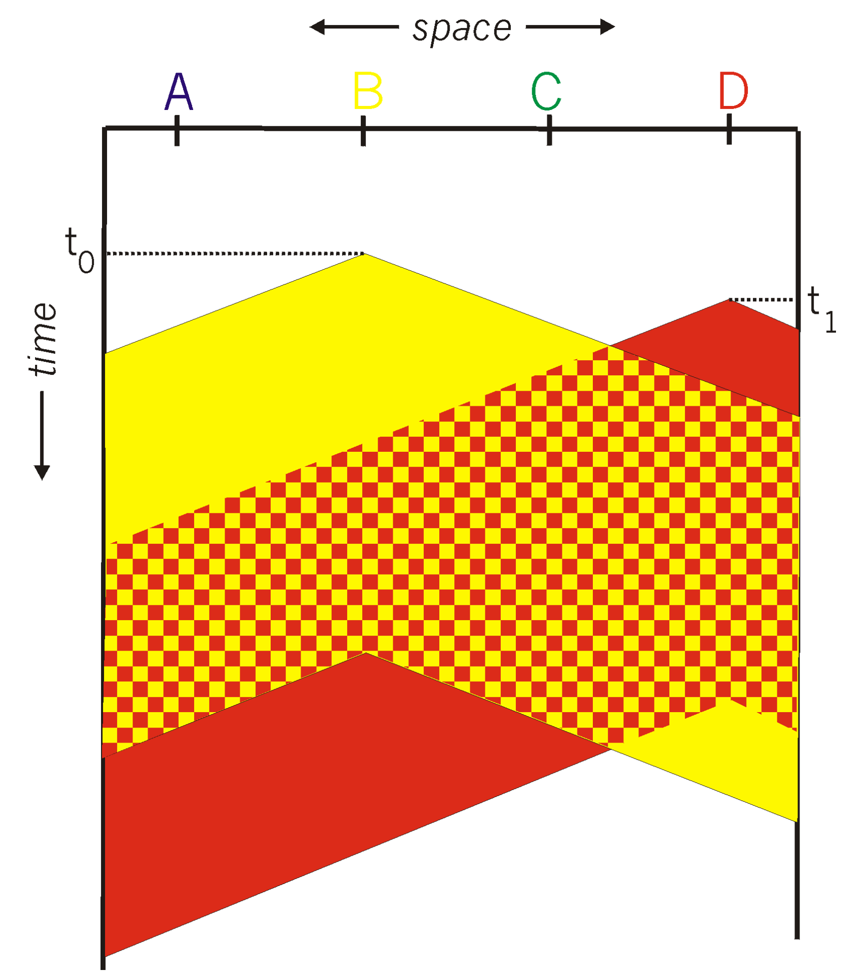
- **When source has a frame to send**
    - Transmit at full bandwidth
    - No a priori coordination among nodes

- **Two or more transmitting sources => collision**
    - Frame lost

- **Link-layer protocol specifies:**
    - How to detect collision
    - How to recover from collisions

CSMA (Carrier Sense Multiple Access)

- CSMA: **listen** before transmit
    - If channel sensed idle: transmit entire frame
    - If channel sensed busy: defer transmission
- Human analogy: don't interrupt others!
- Does this eliminate all collisions?
    - **No**, because of nonzero propagation delay
- Solution:
    - Include a **Collision Detection (CD)** mechanism
    - If a collision detected
        - Retransmit



CSMA/CD (Carrier Sense Multiple Access, Collision Detection)

- CSMA/CD: carrier sensing
  - **Collisions detected within short time**
  - Colliding transmissions aborted, reducing wastage
- Collision detection easy in wired (broadcast) LANs
  - Compare transmitted and received signals
- Collision detection difficult in wireless LANs

Once a collision is detected …

- **When should the frame be resent?**

- Immediately?
  - Every NIC would start sending immediately
  - Collision again!

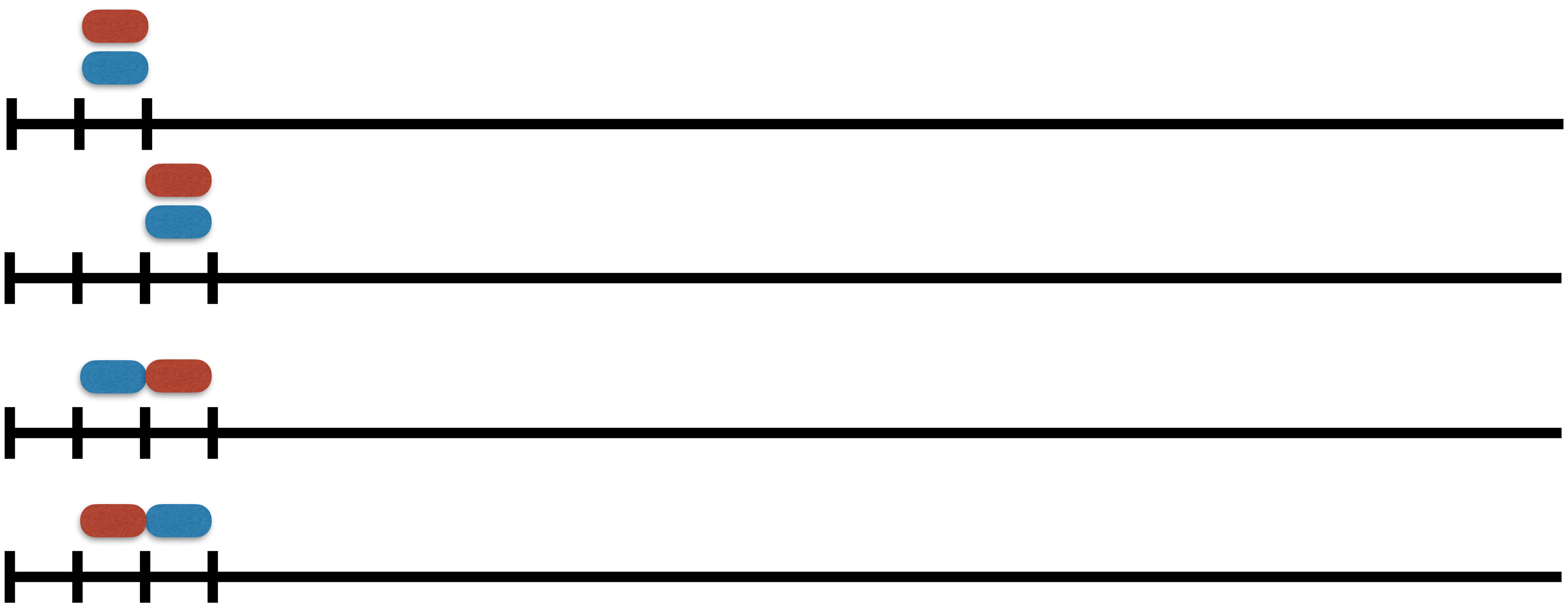
- Take turns?
  - Back to time division multiplexing

CSMA/CD in one slide!

- **Carrier Sense: continuously listen to the channel**
  - If idle: start transmitting
  - If busy: wait until idle
- **Collision Detection: listen while transmitting**
  - No collision: transmission complete
  - Collision: abort transmission; send jam signal
- **Random access: exponential back off**
  - After collision, transmit after “waiting time”
  - After k collisions, choose “waiting time” from {0, …, 2^k-1)
  - Exponentially increasing waiting times
  - But also, exponentially larger success probability

CSMA/CD (Collision Detection): An example

Attempt 1: Suppose a collision happens



Attempt 2: Four possibilities

Success with Probability = 0.5

What is the success probability in attempt 3?

Answer: **0.75**

Why Frames?

(Layering: Link Layer on top of Physical Layer)

Building Link Layer on top of Physical Layer

- Physical layer sends/receives bits on a link, and forwards to link layer
- View at the destination side physical layer:

  010101100111111011111011111100101000111

- Challenge: how to take the above bits and convert to:

  010101100111111011111011111100101000111

- **Problem**: how does the link layer separate data into correct "chunks"?
  - Chunks belonging to different applications
- Data link layer **interfaces** with **physical layer** using **frames**
  - Implemented by the network adaptor
  - **Finally: What are these frames?**

Frames

Identifying start/end of frames: Sentinel Bits

- Delineate frame with special "sentinel" bit pattern
  - e.g., **01111110** -> start, **01111111** -> end

01111110	**Frame contents**	**01111111**

- **Problem: what if the sentinel occurs within the frame?**

- Solution: **bit stuffing**
  - Sender always inserts a **0** after five **1**s in the frame content
  - Receiver always removes a **0** appearing after five **1**s

When Receiver sees five 1s…

01111110	Frame contents	01111111

- If next bit is 0, remove it, and begin counting again
  - Because this must be a stuffed bit
  - we can’t be at beginning/end of frame (those had six/seven 1s)

- If next bit is 1 (i.e., we have six 1s) then:
  - If following bit is 0, this is the start of the frame
    - Because the receiver has seen 01111110
  - If following bit is 1, this is the end of the frame
    - Because the receiver has seen 01111111

Example: Sentinel Bits

- Original data, including start/end of frame:

01111110011111101111110111110010111111

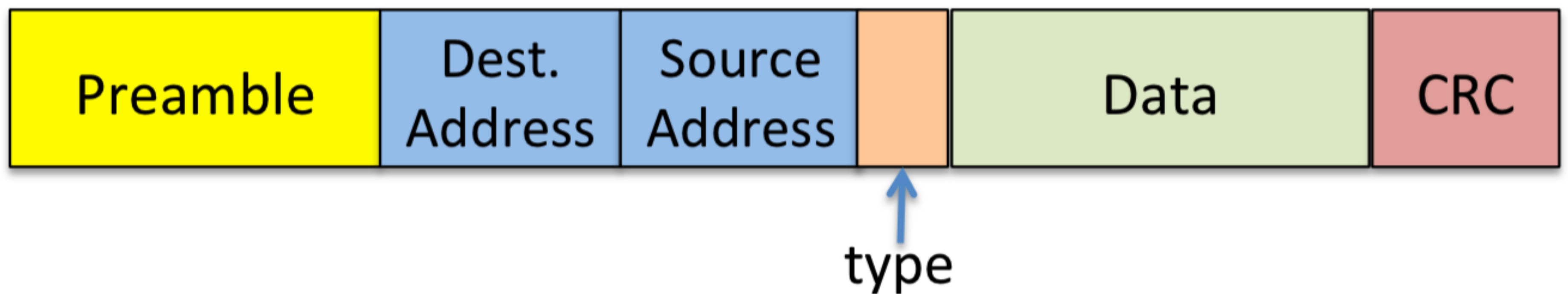
- Sender rule: five 1s -> insert a 0
- After bit stuffing at the sender:

011111100111110101111100111110001 01111111

- Receiver rule: five 1s and next bit 0 -> remove 0

01111110011111101111110111110010111111

Ethernet "Frames"



- **Preamble:**
    - 7 bytes for clock synchronization
    - 1 byte to indicate start of the frame
- **Names:** 6 + 6 bytes (MAC names/addresses)
- **Protocol type:** 2 bytes, indicating higher layer protocol (e.g., IP)
- **Data payload:** max 1500 bytes, minimum 46 bytes
- **CRC:** 4 bytes for error detection

What about source/destination Addresses?

- **Frames are at Layer-2**
  - **Thus, use Layer-2 addresses (MAC names/addresses)**

- **MAC address**
  - Numerical address associated with the network adapter
  - Flat namespace of 6 bytes (e.g., 00-15-C5-49-04-A9 in HEX)
  - Unique, hard coded in the adapter when it is built

- **Hierarchical Allocation**
  - Blocks: assigned to vendors (e.g., Dell) by IEEE
    - First 24 bits (e.g., 00-15-C5-**-**-**)
  - Adapter: assigned by the vendor from its block
    - Last 24 bits

Questions?

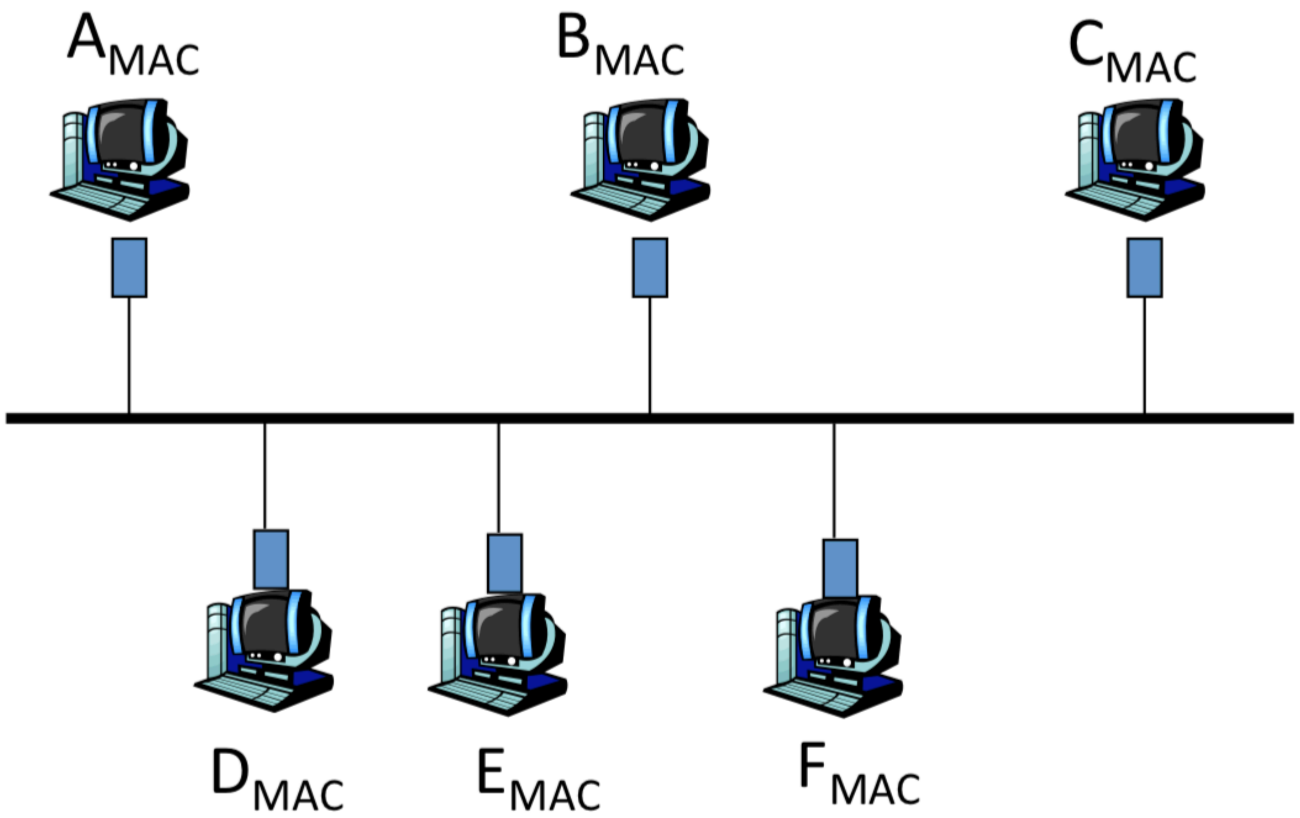
Putting it all together

(Traditional Ethernet)

Traditional Ethernet

- **(Source) Link layer receives data from the network layer (more later)**
- **(Source) Link layer divides data into frames**
  - How does it know source/destination MAC names?
  - Source name is easy … destination name is tricky (more later)
- **(Source) Link layer passes the frame to physical layer**
  - Frames up the frames (using sentinel bits)
  - **And broadcasts on the broadcast Ethernet**
- **(EACH) physical layer regenerates the frame…**
  - And sends it up to the (destination) link layer ….
    - Which sends the data to the network layer …. **If and only if:**
      - destination name matches the receiver's MAC name
      - Or, the destination name is the broadcast address (FF:FF:FF:FF:FF:FF)

Traditional Ethernet



- Ethernet is "plug-n'play"
  - A new host plugs into the Ethernet is good to go
  - No configuration by users or network operators
  - Broadcast as a means of bootstrapping communication

Performance of CSMA/CD

- **Time spent transmitting a frame (collision)**
  - Proportional to distance d; why?
- **Time spent transmitting a frame (no collision)**
  - Frame length p divided by bandwidth b
- **Rough estimate for efficiency (K some constant)**

$$E \sim \frac{\frac{p}{b}}{\frac{p}{b} + Kd}$$

- Observations:
  - For large frames AND small distances, E ~ 1
  - **Right frame length depends on b, K, d**
  - **As bandwidth increases, E decreases**
    - That is why high-speed LANs are switched

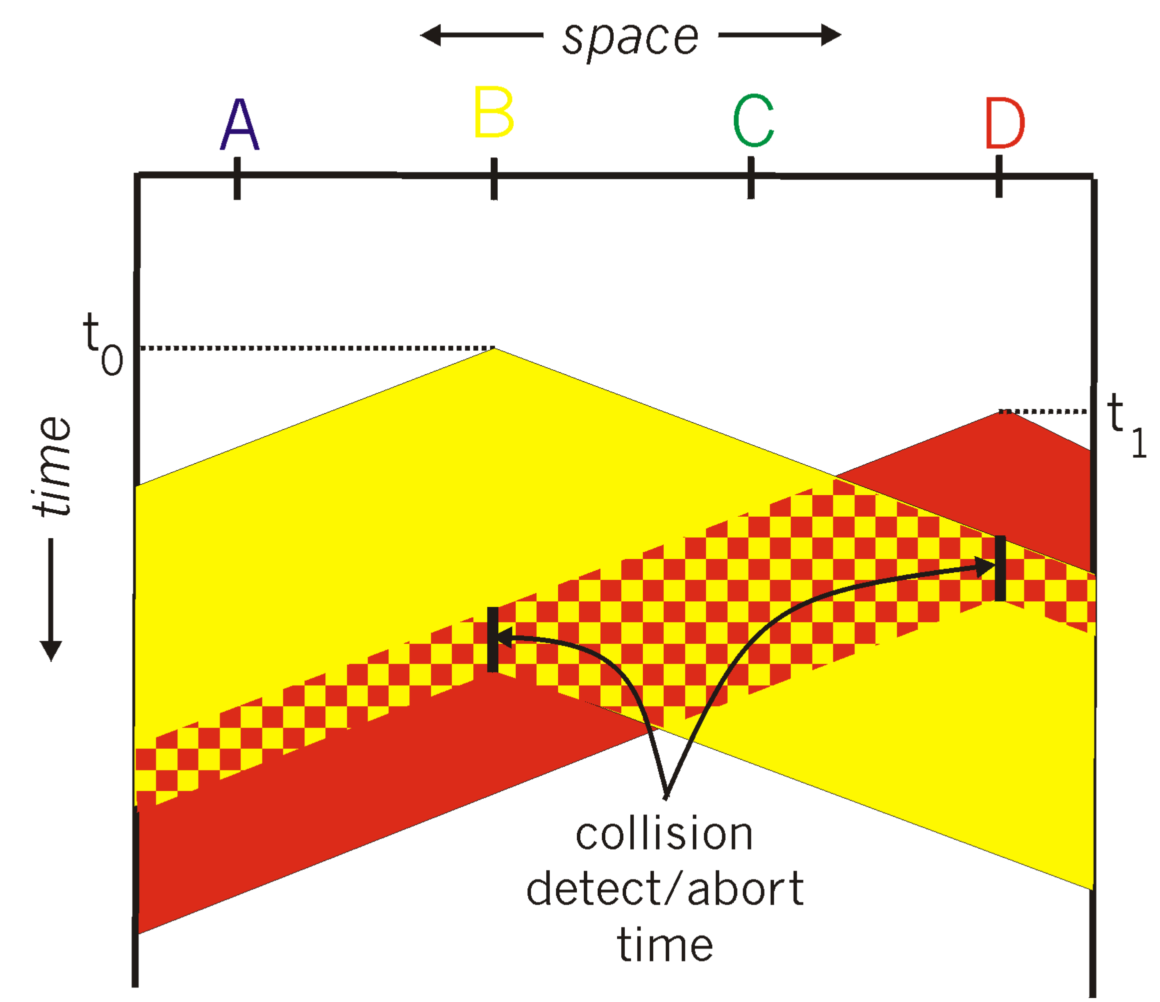
Questions?

CSMA/CD (Collision Detection)

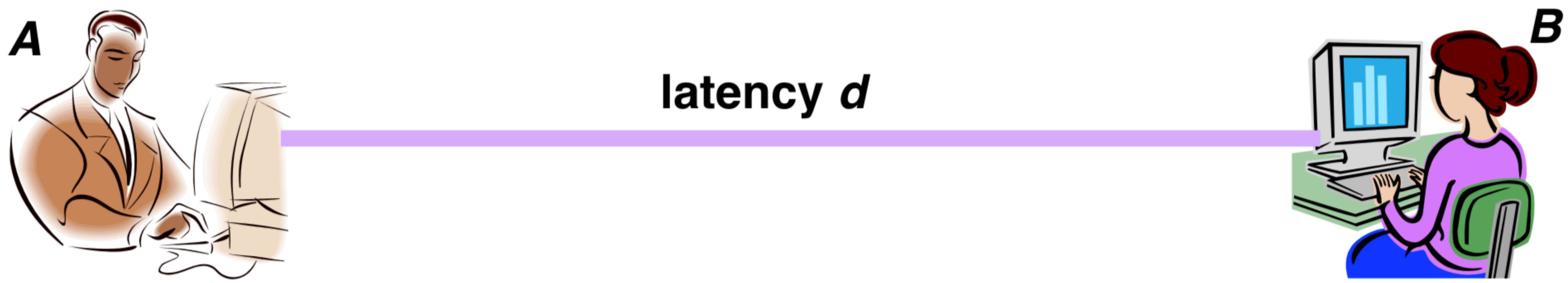
- B and D can tell that collision occurred

- However, need restrictions on
    - **Minimum frame size**
    - **Maximum distance**

- **Why?**

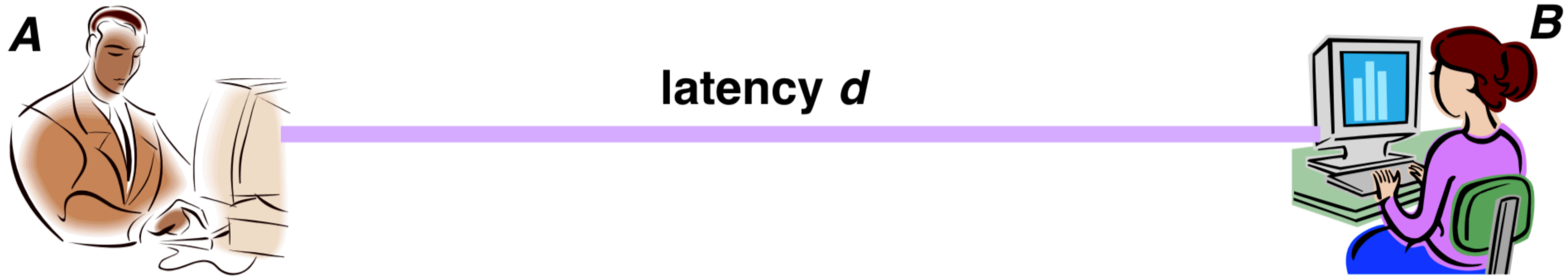


Limits on CSMA/CD Network Length and Frame Size



- **Latency depends on physical length of link**
  - Time to propagate a bit from one end to the other
- **Suppose A sends a packet at time 0**
  - B sees an idle line at all times before **d**
  - … so B happily starts transmitting a packet
- **B detects a collision at time d, and sends jamming signal**
  - But A can't see collision until **2d**
  - **A must have a frame size such that transmission time > 2d**
  - **Need transmission time > 2 * propagation delay**

Limits on CSMA/CD Network Length and Frame Size



- **Transmission time > 2 * propagation delay**
- **Imposes restrictions.**
  - **Example: consider 100 Mbps Ethernet**
  - **Suppose** minimum frame length: 512 bits (64 bytes)
    - Transmission time = 5.12 μsec
    - Thus, we want propagation delay < 2.56 μsec
    - Length < 2.56 μsec * speed of light
    - Length < 768m
- **What about 10Gbps Ethernet?**

Performance of CSMA/CD

- **Time spent transmitting a frame (collision)**
  - Proportional to distance d; why?
- **Time spent transmitting a frame (no collision)**
  - Frame length p divided by bandwidth b
- **Rough estimate for efficiency (K some constant)**

$$E \sim \frac{\frac{p}{b}}{\frac{p}{b} + Kd}$$

- Observations:
  - For large frames AND small distances, E ~ 1
  - **Right frame length depends on b, K, d**
  - **As bandwidth increases, E decreases**
    - That is why high-speed LANs are switched

Evolution

- **Ethernet was invented as a broadcast technology**
  - Hosts share channel
  - Each packet received by all attached hosts
  - CSMA/CD
- **Current Ethernets are "switched" (next lecture)**
  - Point-to-point medium between switches;
  - Point-to-point medium between each host and switch
  - No sharing, no CSMA/CD