CS4450

Computer Networks:
Architecture and Protocols

Lecture 6
Data Link Layer

Rachit Agarwal

Announcements

- We will have a “live” coding class on 02/28
    - **Please bring your laptops**
    - Its gonna be a lot of fun!
    - We will learn how to implement sockets, etc.

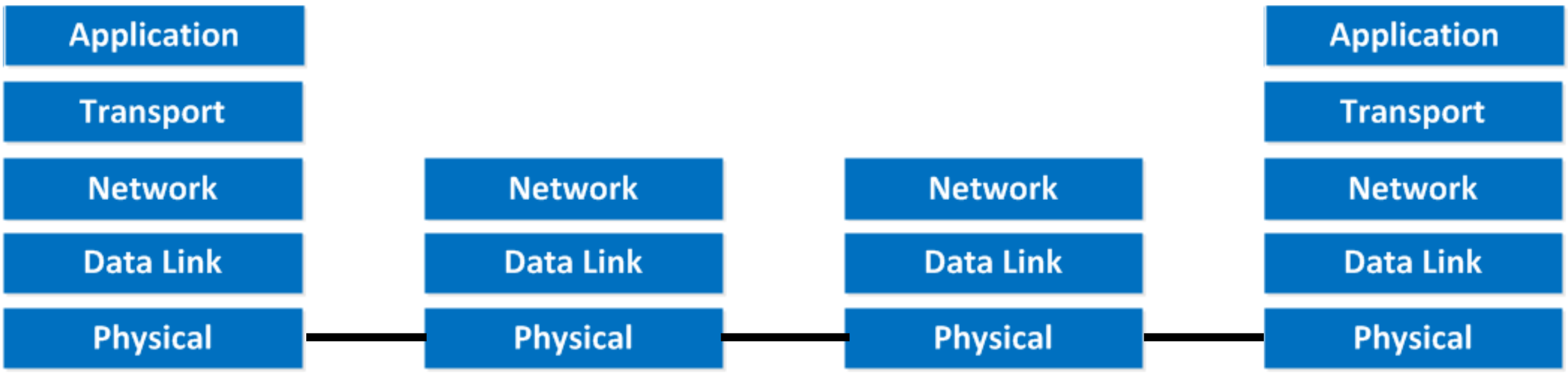
- Please read Chapter 1 of the textbook!
    - And try to solve problems at the end of Chapter 1 of the textbook
        - For extra practice
        - Ask us questions on Piazza

- **In-class Quiz policy**
    - **We understand that sometimes its impossible to attend a lecture**
    - **Email AT LEAST AN HOUR BEFORE the lecture if you can’t attend**
    - If legitimate reasons, we will ignore that quiz for you

Quick recap from last lecture

Recap: Three design principles

- How to break system into modules
  - **Layering**
- Where are modules implemented
  - **End-to-End Principle**
- Where is state stored?
  - **Fate-Sharing**



Recap: Internet Design Goals

- **Build something that works**
- Connect existing networks
- Robust in face of failures
- Support multiple types of delivery service
- Accommodate a variety of networks
- Allow distributed management
- Easy host attachment
- Cost effective
- Allow resource accountability

Context for Today's Lecture

- You now understand
    - Network sharing (in depth)
    - Architectural principles and design goals (in depth)
    - End-to-end working of the Internet (at a high-level)

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

- **Today: 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

- **Communication Medium**
  - **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!

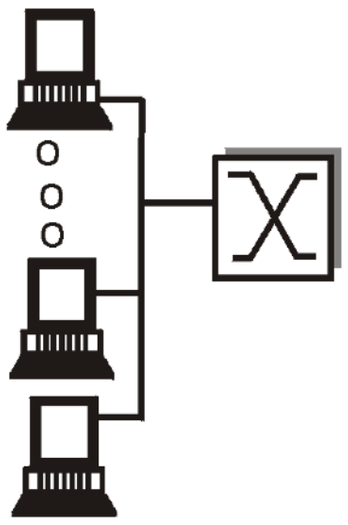
- **Network Adapters (e.g., NIC — network interface card)**
  - The hardware that connects a machine to the network
  - Has a “name” — MAC (Medium access control) address



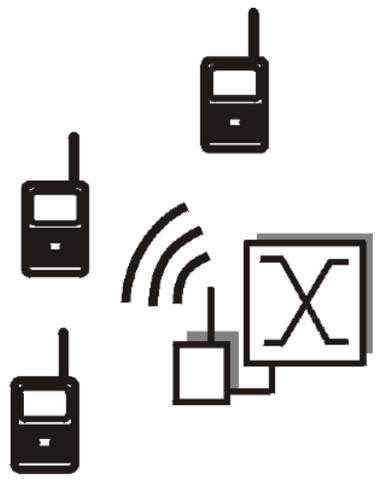
Point-to-Point vs. Broadcast Medium

- Point-to-point: **dedicated** pairwise communication
    - E.g., long distance fiber link
    - E.g., Point-to-point link between two routers

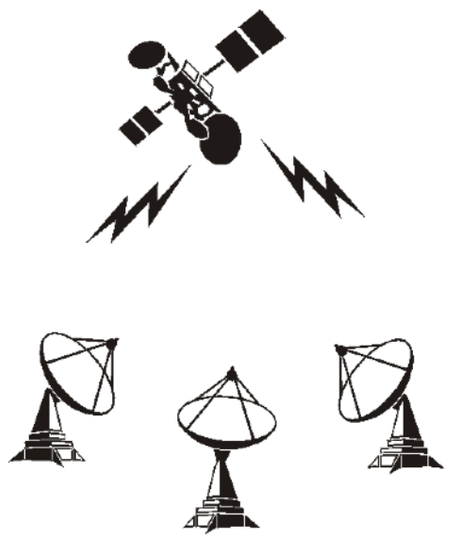
- Broadcast: **shared** wire or medium
    - Traditional Link Layer (Ethernet)
    - 802.11 wireless LAN



shared wire
(e.g. Ethernet)



shared wireless
(e.g. Wavelan)



satellite

cocktail party

Data Link Layer: Broadcast (until ~2000s)

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

Broadcast Medium: Desirable properties

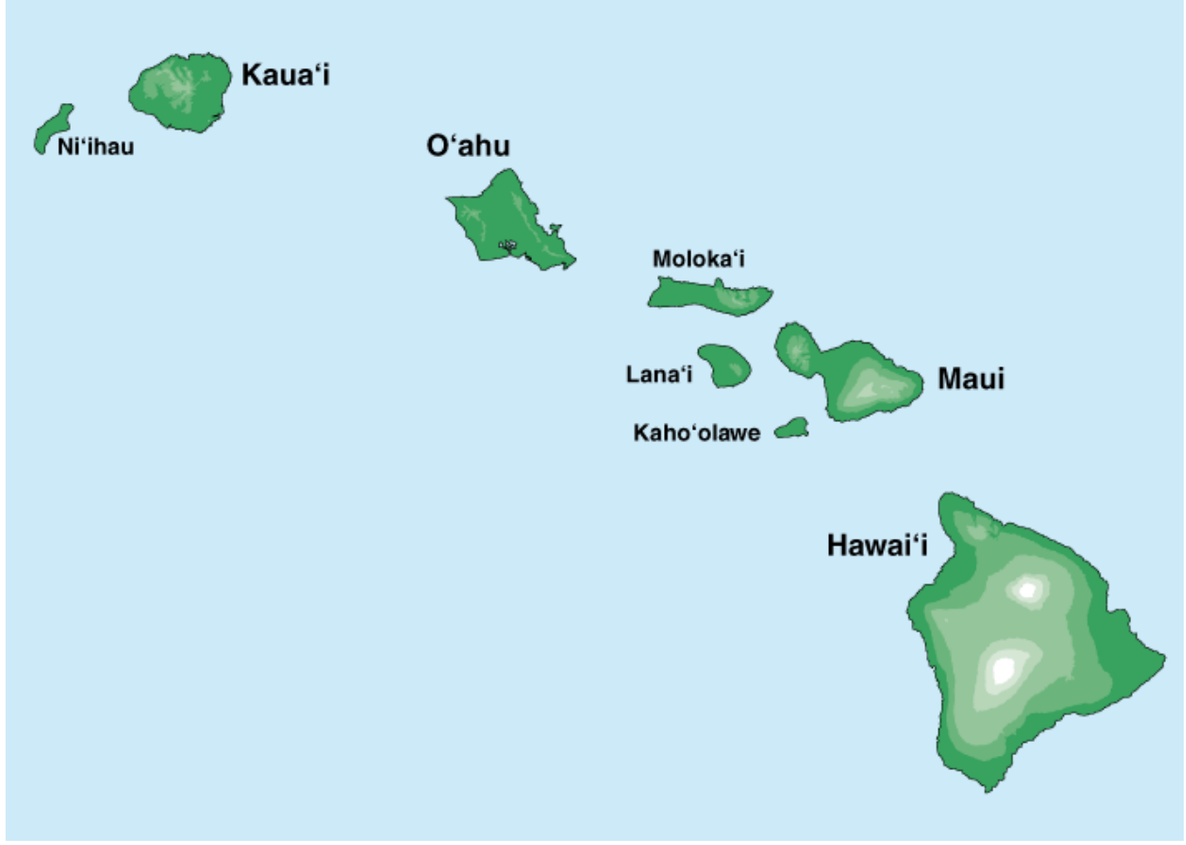
- **One and only one: data delivery**

- How do we design a broadcast medium protocol for data delivery?

link-layer “protocol”

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

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 wireless networks
    - E.g., radio

Time-Division Multiple Access (TDMA)

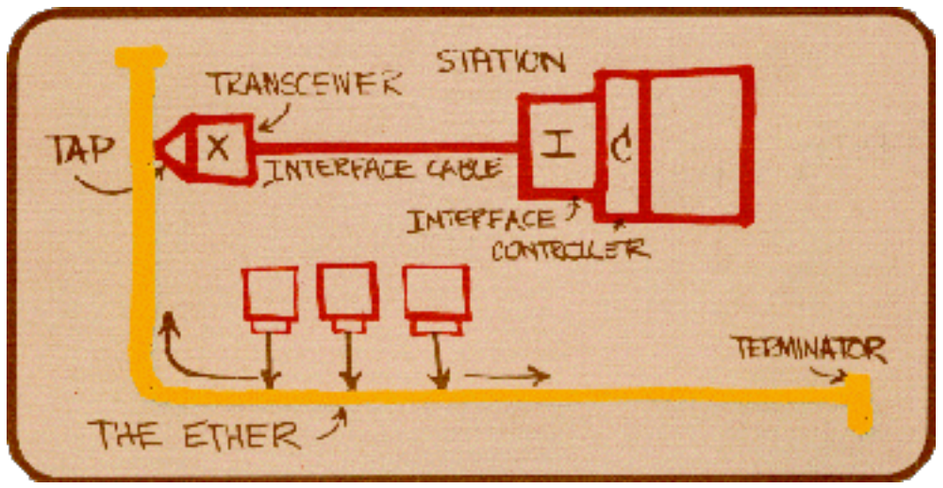
- **Time sharing**
    - Divide time into **slots**
    - Divide data into **frames**
        - Such that a frame can be transmitted in one slot
    - **Every source is assigned a subset of slots**
        - And transmits a frame only in its assigned slot

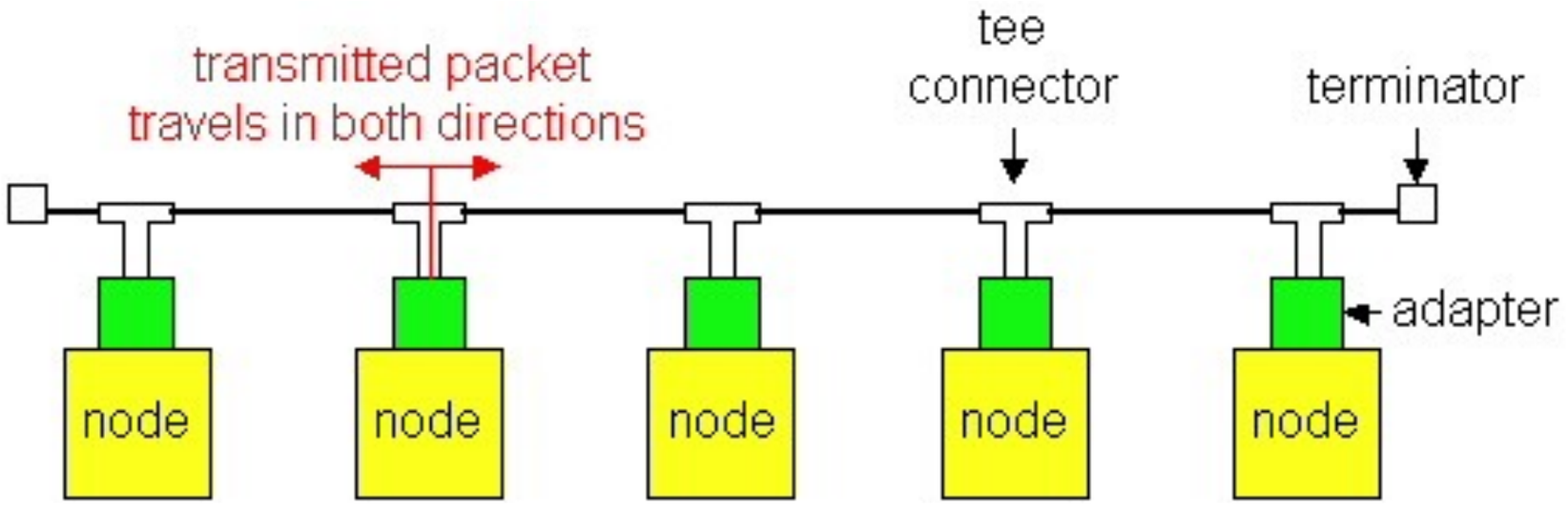
- **Goods: no collisions**

- **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 (MAC) Protocol

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

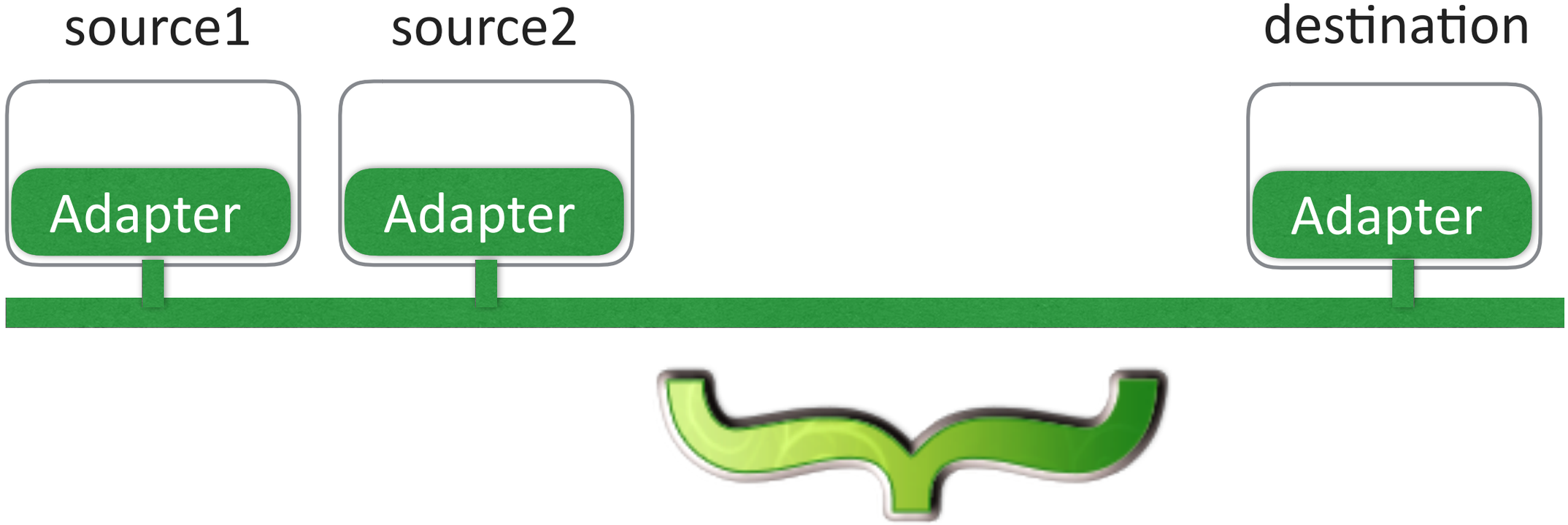
- **Two or more transmitting sources => collision**
    - Frame lost

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

LETS TRY!

Group Exercise:

Multiple source-destination pairs
Design a protocol that allows sharing the broadcast medium



link-layer “protocol”

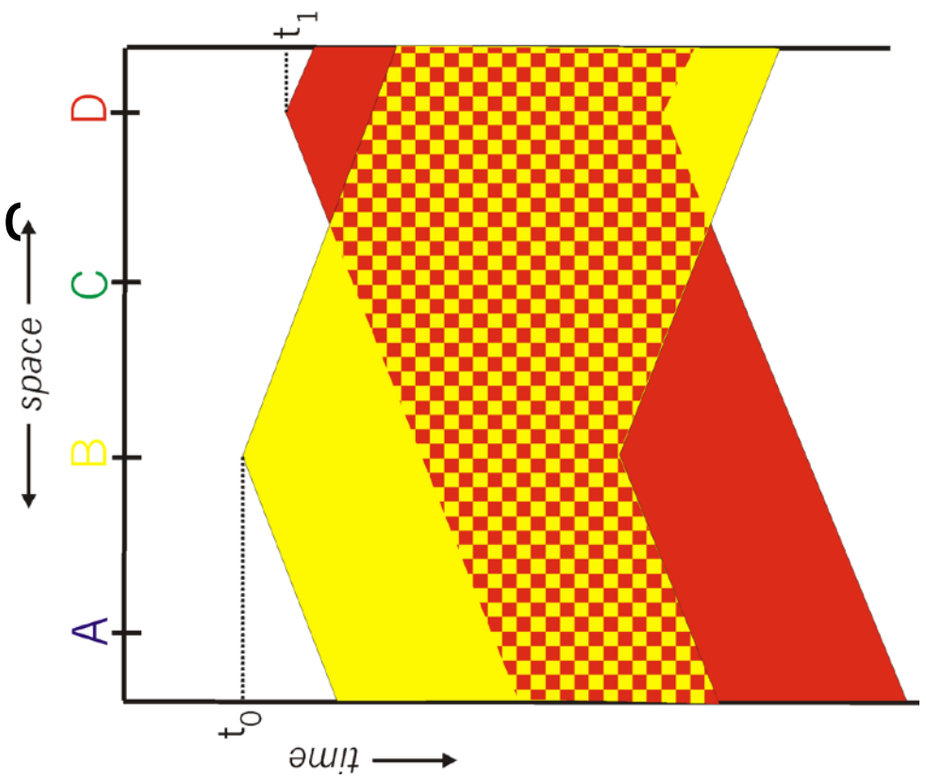
CSMA (Carrier Sense Multiple Access)

- 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)** mec
  - 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

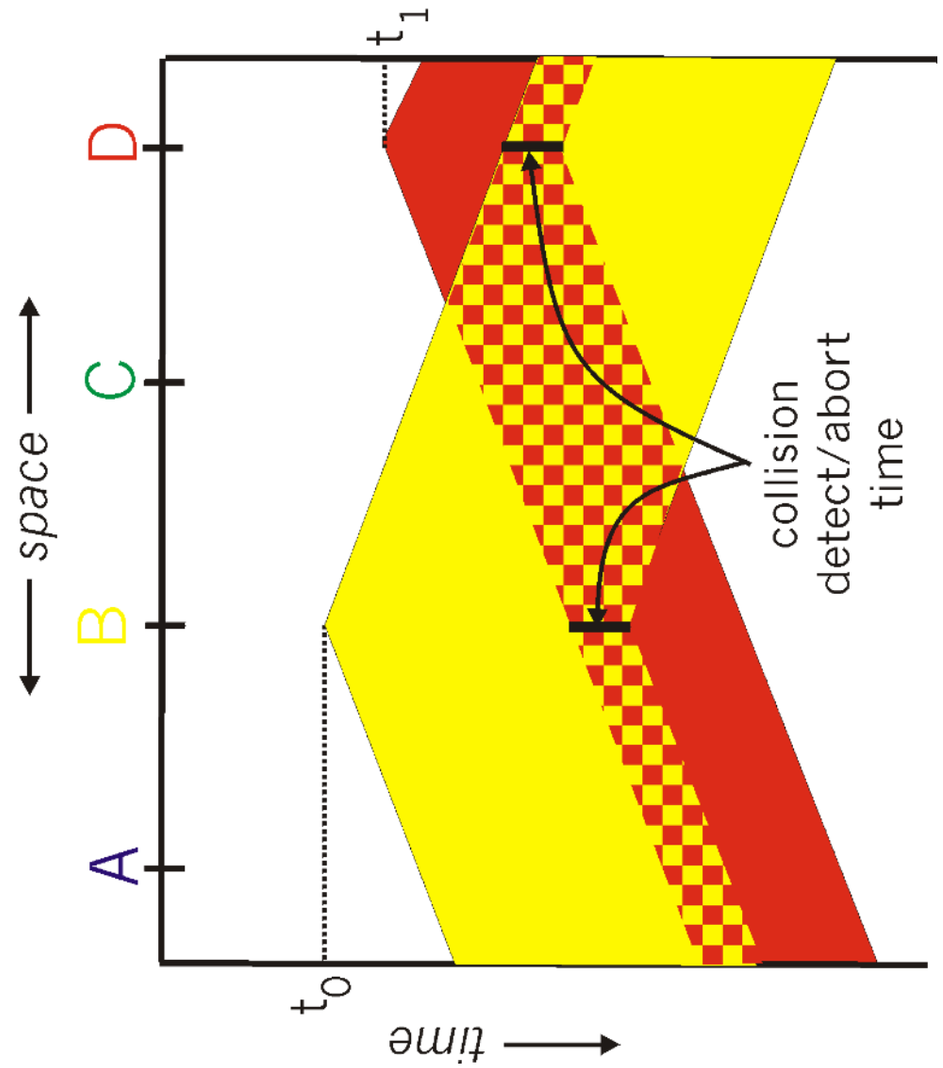
- Collision detection difficult in wireless LANs

CSMA/CD (Collision Detection)

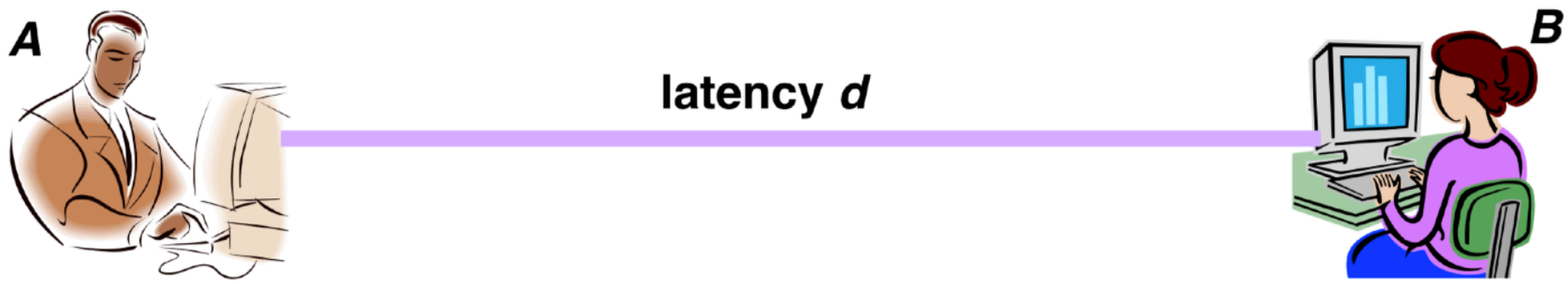
- **B** and **D** can tell that collision occurred

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

- **Why?**



Limits on CSMA/CD Network Length and Frame Size

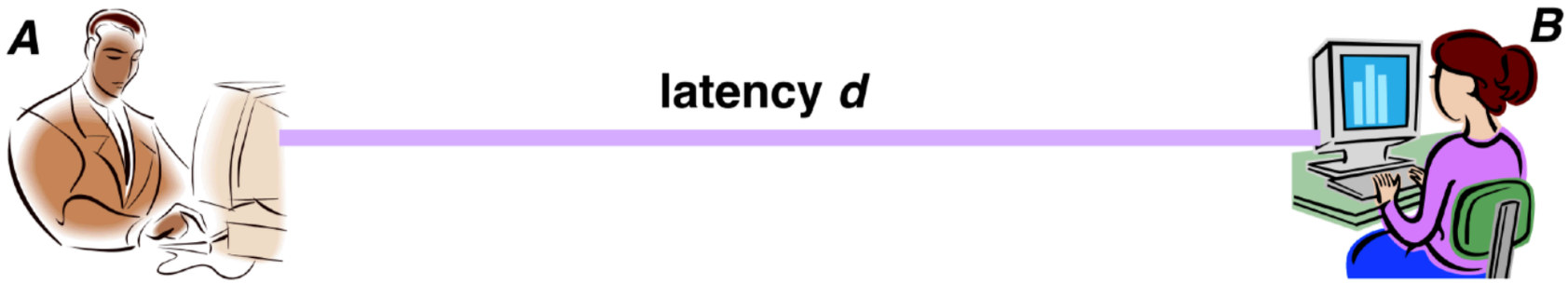


- **Latency depends on physical length of link**
    - Time to propagate a bit from one end to the other

- **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?**

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

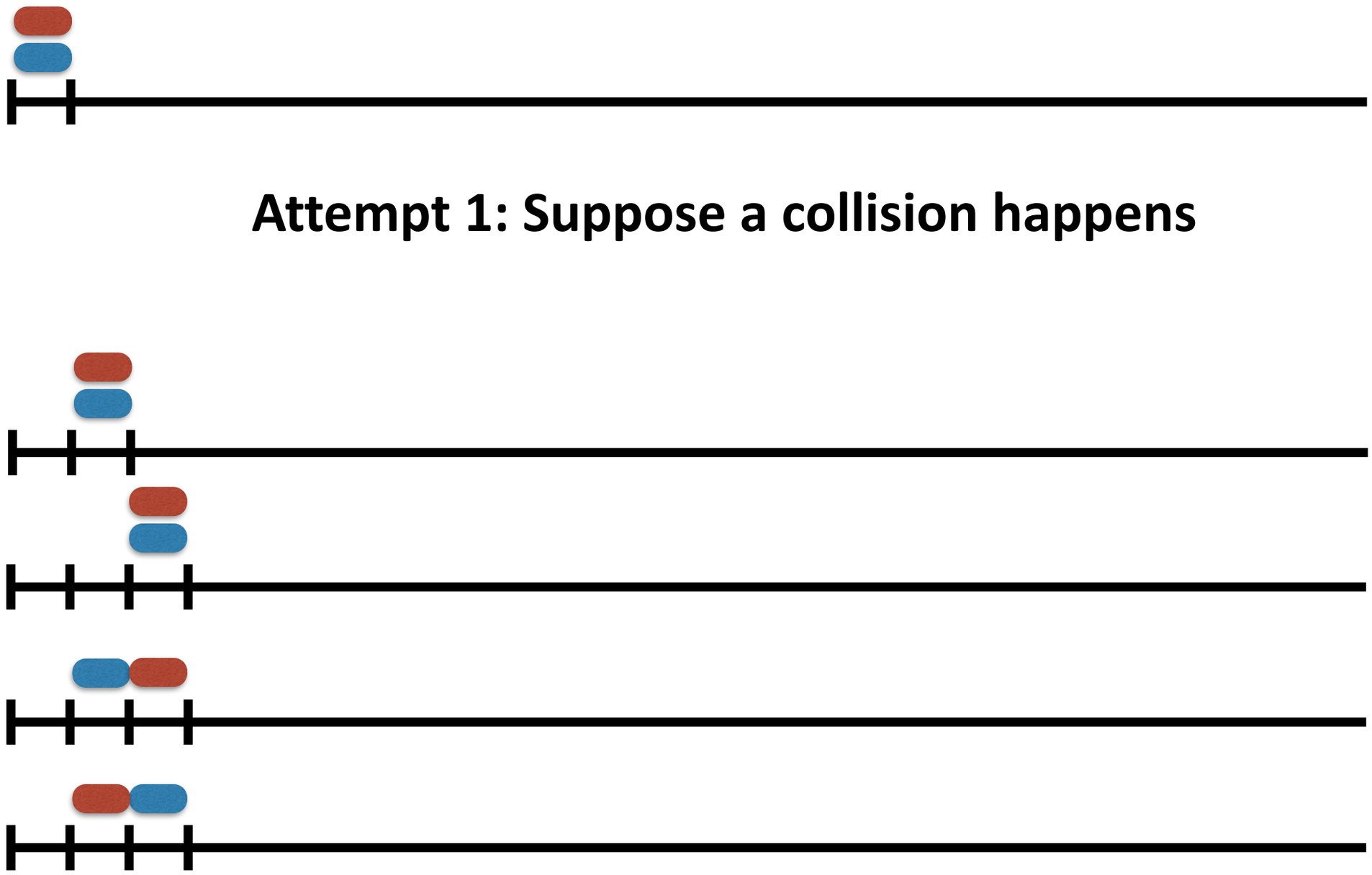
CSMA/CD in one slide!

- **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

Group Exercise:

What is the success probability in attempt 3?

Answer: **0.75**

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