CS/ECE 4457

Computer Networks:
Architecture and Protocols

Lecture 6
Data Link Layer

Qizhe Cai

Announcements

- **Exam 1 on 02/10**
  - **Material: everything covered until lecture 7**
    - **Slides, Problem set 1, Problem set 2 (up to Question 4)**
  - Should be doable in ~90 minutes
  - Open-notes, open-book, except…
  - Talking to any human or alien

- **Exam structure**
  - Several conceptual questions
  - Several "problems" (e.g., Q1 and Q2 on pset 2)

- **For all those who declared their conflicts**
  - We have already sent an email; please respond by tomorrow
  - If we missed you, meet me after the lecture today

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)

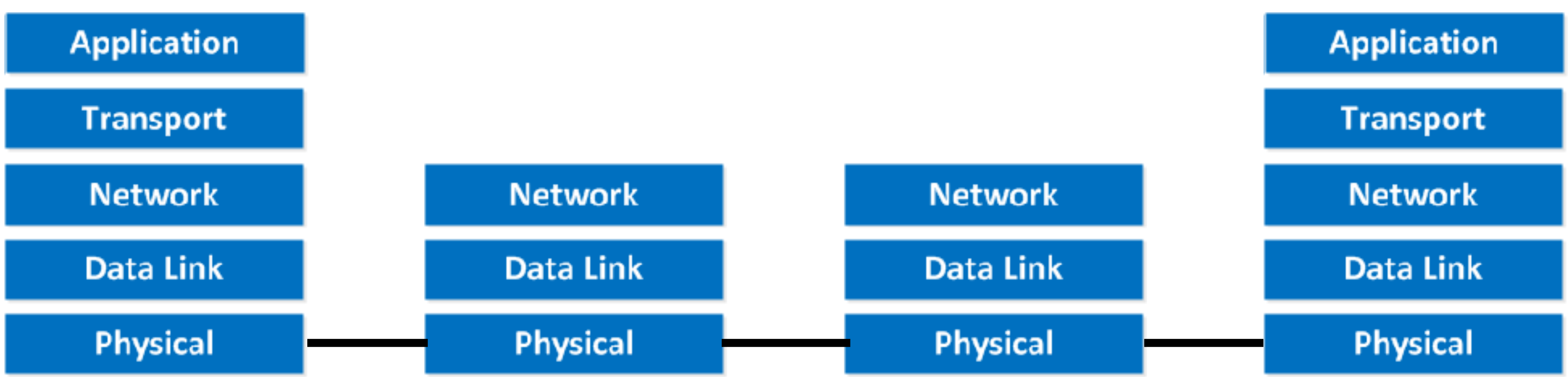
- Now, time to dive deeper:
  - Link Layer (~1 week)
  - Network Layer (~4 weeks)
  - Transport Layer (~3 weeks)

- **Today: Link layer**

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**



From Architecture to Design:

Design Goals

David Clark

- Wrote a paper in 1988 that tried to capture why the Internet turned out as it did
- It described an ordered list of priorities that informed the decision
- What do you think those priorities were?

#1: Connect Existing Networks

Want one protocol that could be used to connect any pair of (existing) networks

- Different networks may have different needs
  - For some: reliable delivery more important
  - For others: performance more important
  - **But there is one need that every network has: connectivity**
- The Internet Protocol (IP) is that unifying protocol
  - All (existing) networks must be able to implement it

#2: Robust in Face of Failures

As long as network is not partitioned, two hosts should be able to communicate (eventually)

- Must **eventually recover** from failures
- Very successful in the past; unclear how relevant now
    - **Availability** is becoming increasingly important than **recovery**

#3: Support Multiple Types of Delivery Services

Different delivery services (applications) should be able to co-exist

- Already implies an application-neutral framework
- Build lowest common denominator service
    - **Again: connectivity**
    - Applications that need reliability may use it
    - Applications that do not need reliability can ignore it

Questions?

#4: Variety of Networks

Must be able to support different networks with different hardware

- **Incredibly successful!**
  - Minimal requirements on networks
  - No need for reliability, in-order, fixed size packets, etc.
  - A result of aiming for lowest common denominator
- **Again: Focus on connectivity**
  - Let networks do specific implementations for other functionalities
  - Automatically adapt: WiFi, LTE, 3G, 4G, 5G ….

#5: Decentralized Management

No need to have a single "vantage" point to manage networks

- Both a curse and a blessing
  - Important for easy deployment
  - Makes management hard today

- Recent efforts have improved management of individual networks
  - But no attempt to manage the Internet as a whole…
  - What might make this complex?

#6: Easy Host Attachment

The mechanism that allows hosts to attach to networks must be made as easy as possible, but no easier

- Clark observes that cost of host attachment may be higher because hosts had to be smart
- But the administrative cost of adding hosts is very low, which is probably more important
    - Plug-and-play kind of behavior…
- And now most hosts are smart for other reasons
    - So the cost is actually minimal…

#7: Cost Effective

Make networks as cheap as possible, but no cheaper

- Cheaper than circuit switching at low end
- More expensive than circuit switching at high end
- Not a bad compromise:
    - Cheap where it counts (low-end)
    - More expensive for those who can pay…

#8: Resource Accountability

Each network element must be made accountable for its resource usage

- Failure!

Internet Motto

“We reject kings, presidents and voting. We believe in rough consensus and running code.”

- - David Clark

Real 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

Questions to think about

- What goals are missing from this list?
    - **Suggestions?**

Some of the missing issues

- **Performance**
- Security
  - Resilience to attacks (denial-of-service)
  - Endpoint security
  - Tracking down misbehaving users
- Privacy
- Availability
- Resource sharing (fairness, etc.)
- ISP-level concerns
  - Economic issues of interconnection

Questions to think about

- What goals are missing from this list?
    - **Suggestions?**

- What would the resulting design look like?

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!

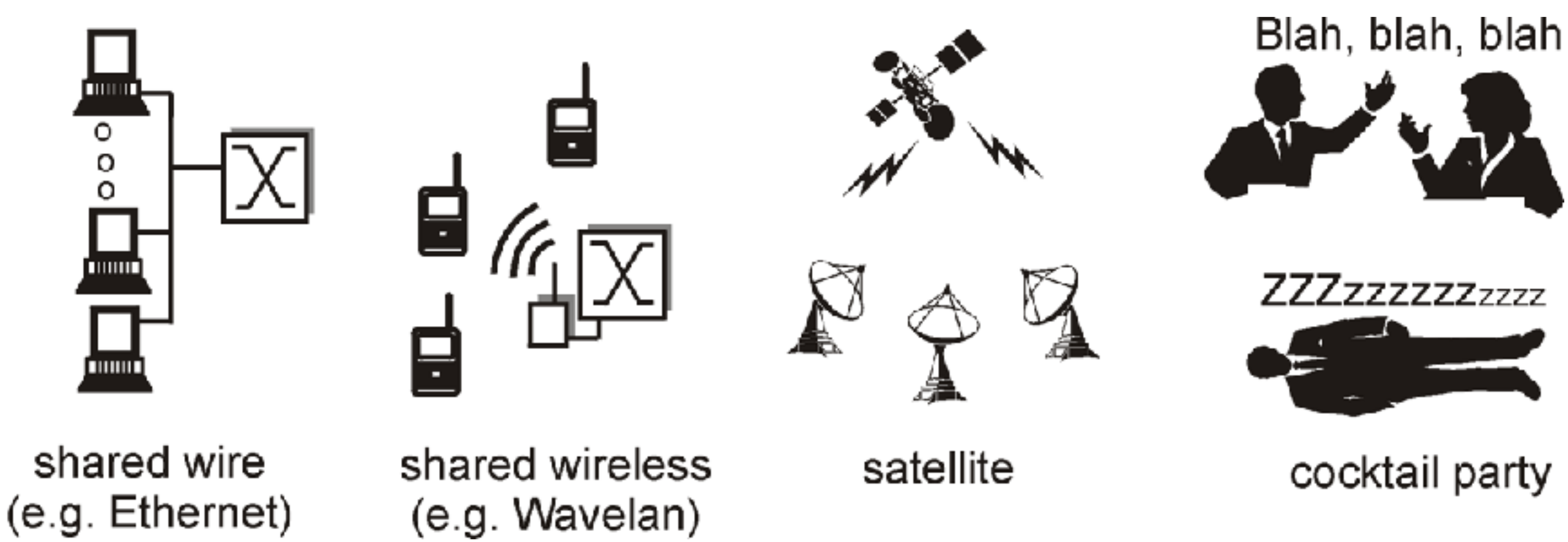
- **Network Adapters (e.g., NIC — network interface card)**
  - The hardware that connects a machine to the network
  - Has a “name” — MAC (Medium access control) address




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

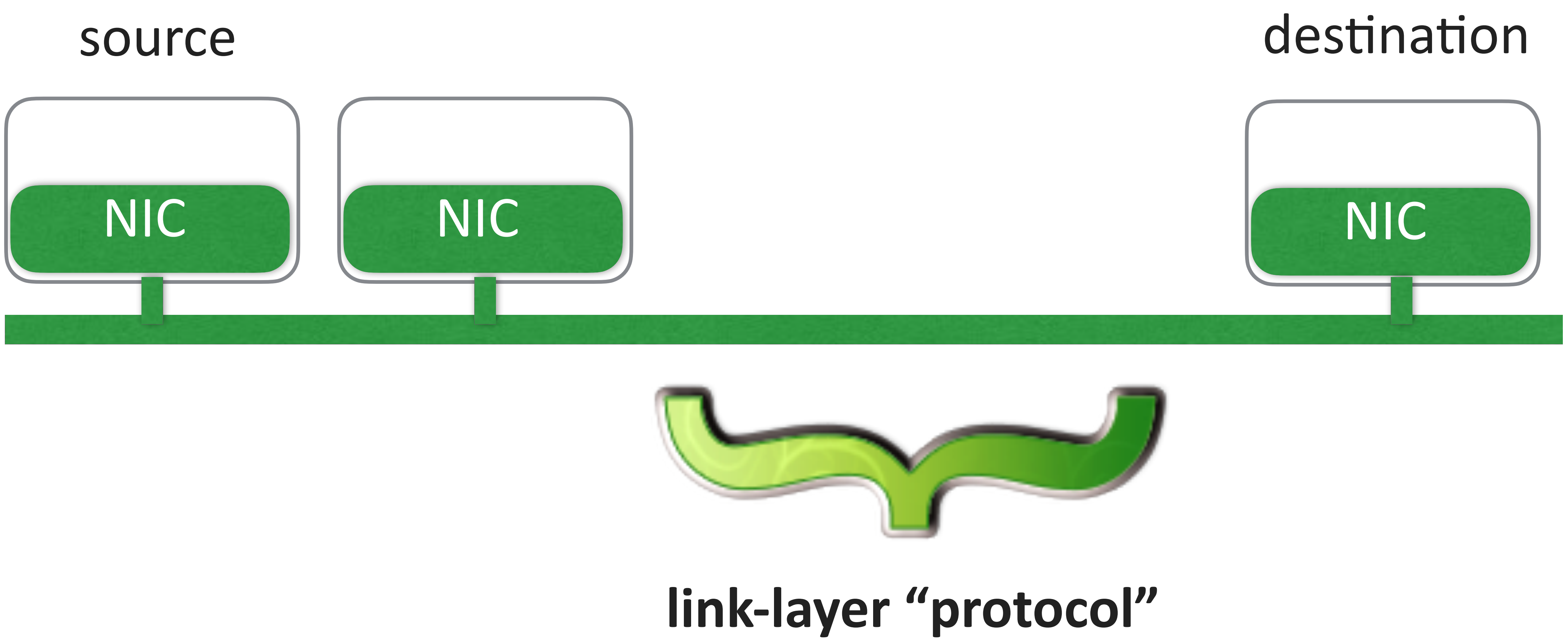


Data Link Layer: Broadcast (until ~2000s)

- **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

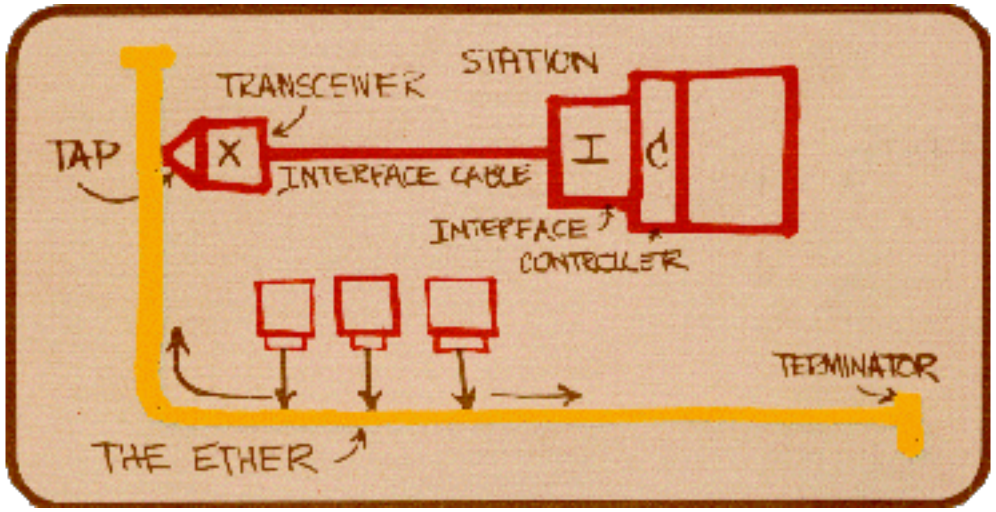
- Used in many wireless networks
  - E.g., radio

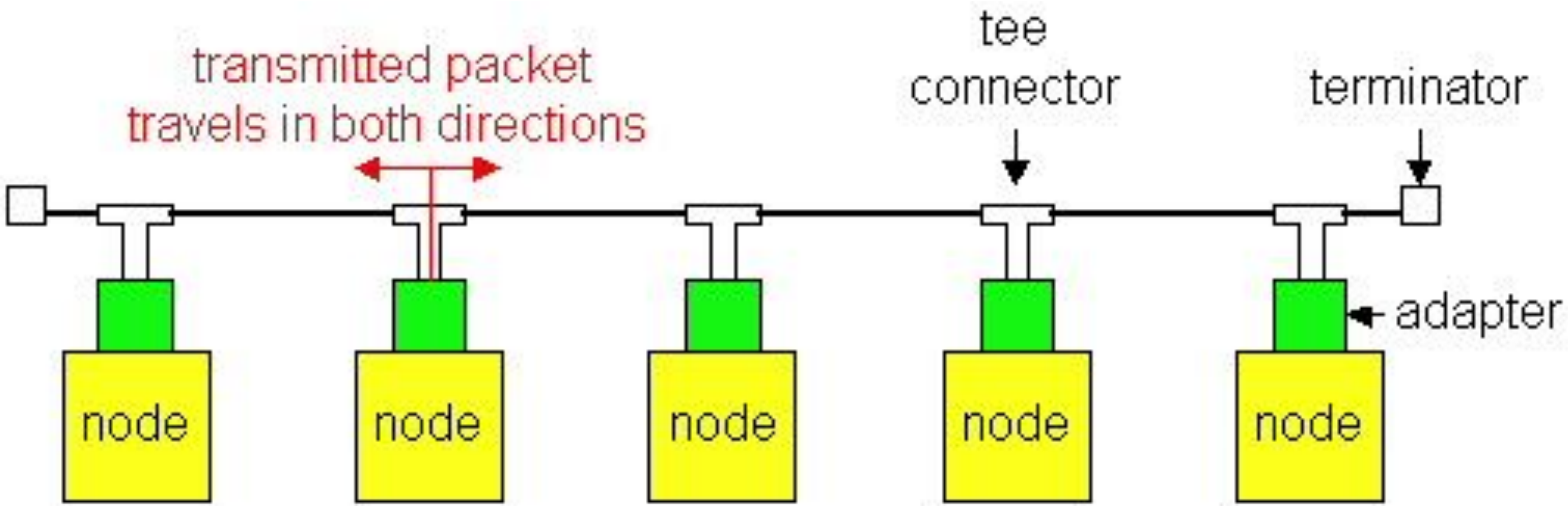
Time-Division Multiple Access (TDMA)

- **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 :-)

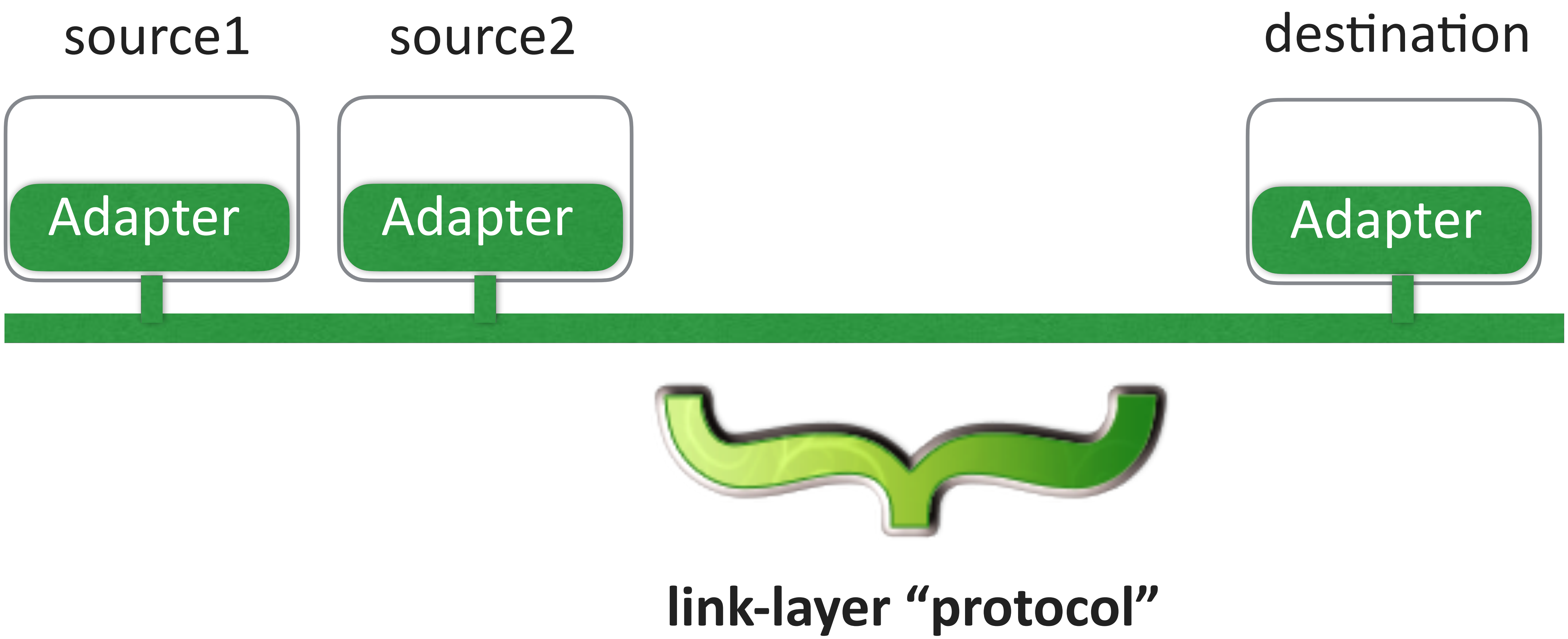
Link Layer (Media Access Control, or MAC) Protocol

- **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

LETS TRY!

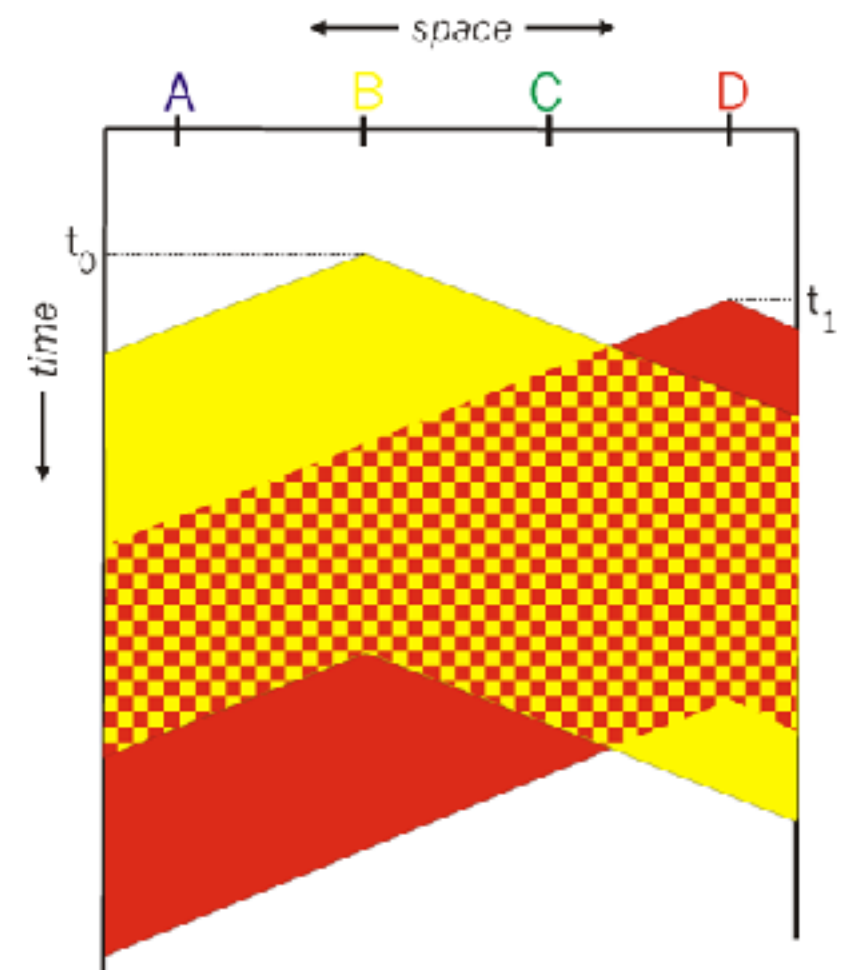
Multiple source-destination pairs

Design a protocol that allows sharing the broadcast medium



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)

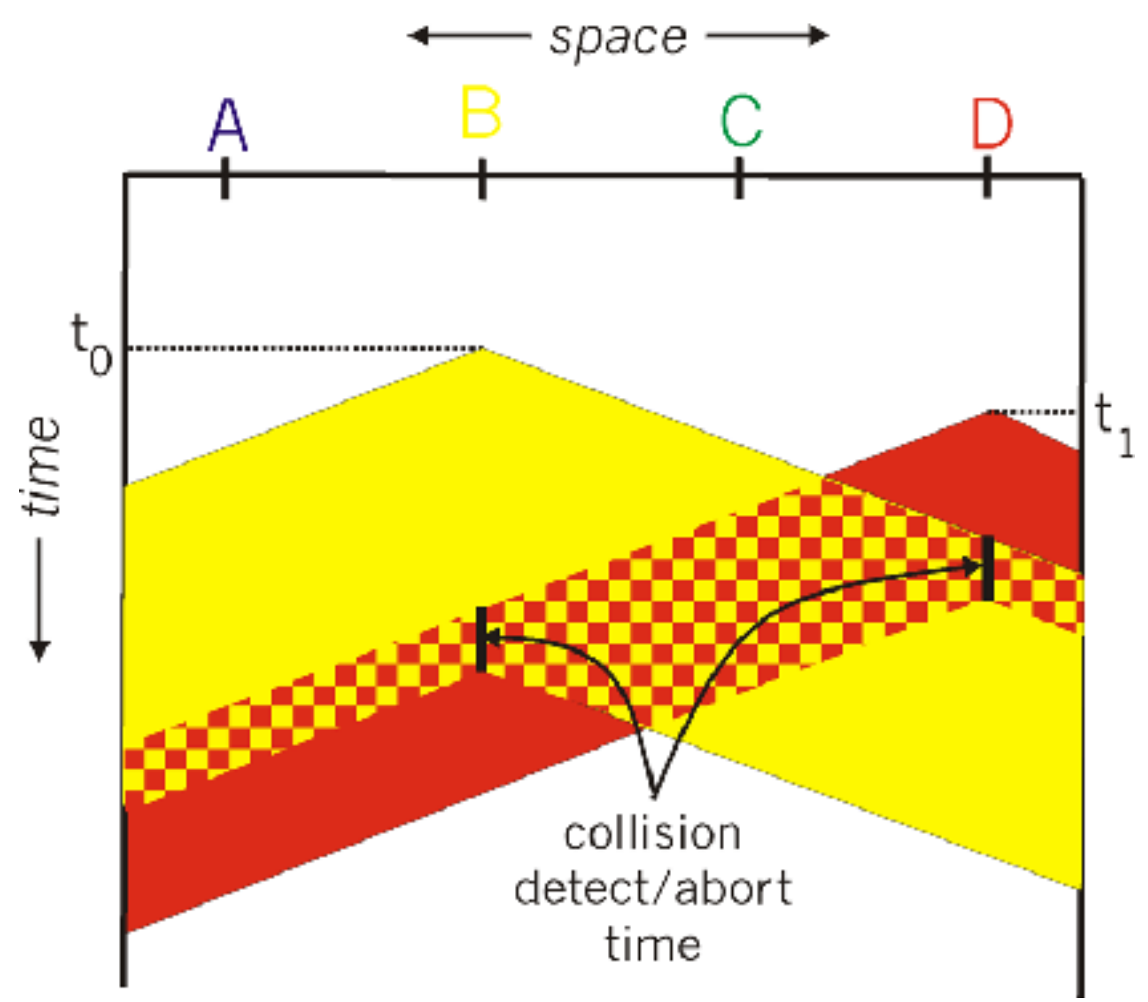
- CSMA/CD: carrier sensing
  - **Collisions detected within short time**
  - Colliding transmissions aborted, reducing wastage
- Collision detection easy in wired (broadcast) LANs
  - Compare transmitted and received signals
- Collision detection difficult in wireless LANs

CSMA/CD (Collision Detection)

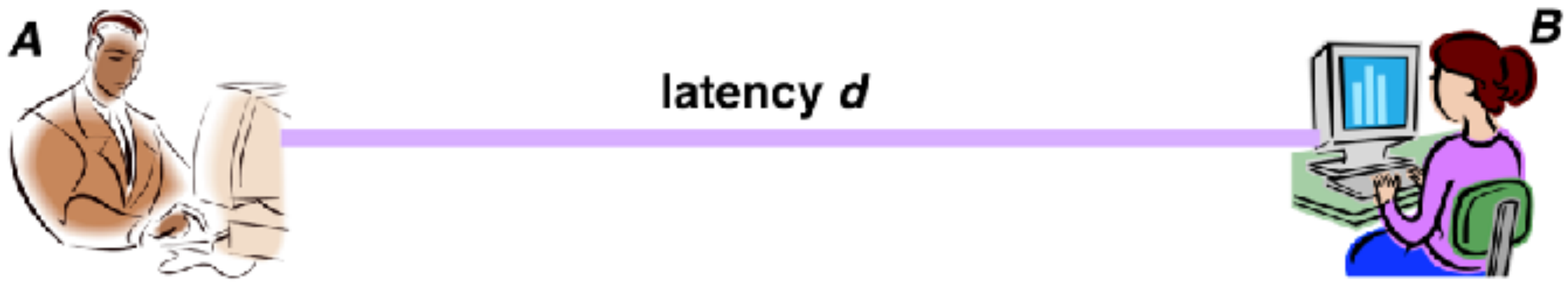
- B and D can tell that collision occurred

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

- **Why?**

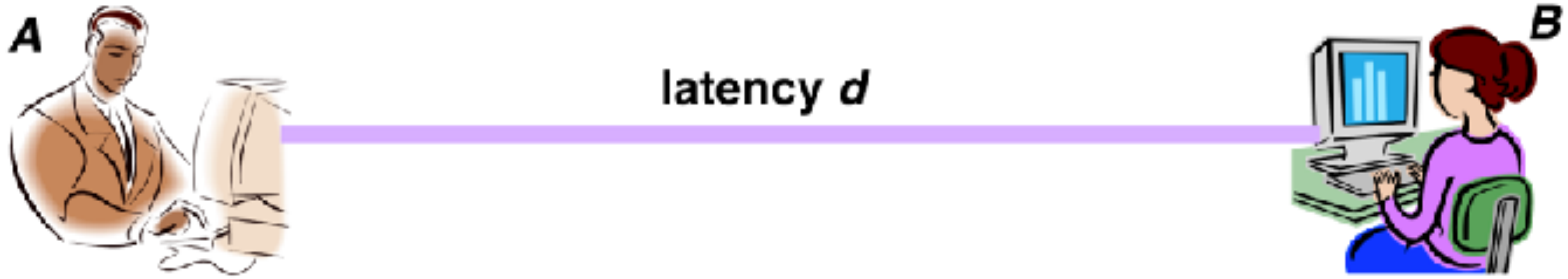


Limits on CSMA/CD Network Length and Frame Size



- **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!

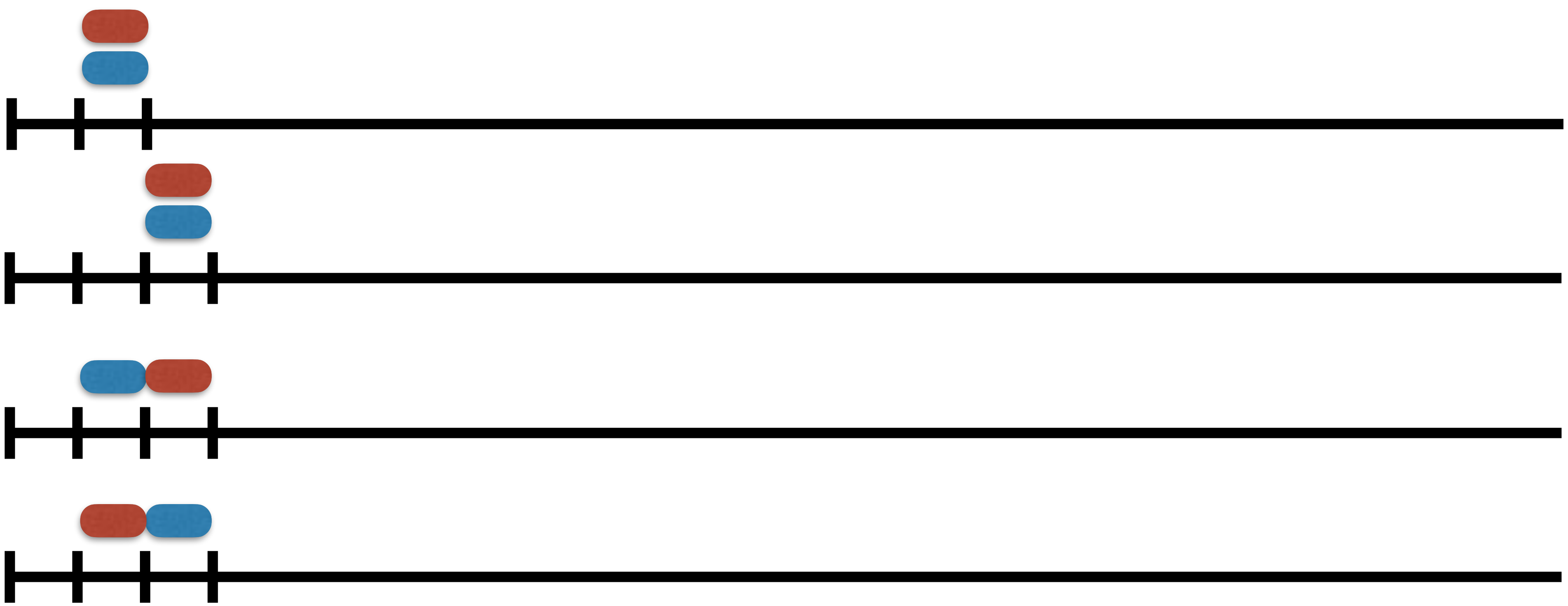
- Take turns?
    - Back to time division multiplexing

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

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 size 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