Application Layer: HTTP (Contd.)

CPSC 433/533, Spring 2021

Anurag Khandelwal

Meeting HTTP Goals

- **Users**
  - Fast downloads (not the same as low-latency communication!)
  - High availability
- **Content provider**
  - Happy users (hence, above)
  - Cost-effective delivery infrastructure
- **Network (secondary)**
  - Avoid overload

1. Improve HTTP to compensate for TCP weak-spots
2. Caching & Replication
3. Exploit economies of scale (Webhosting, CDNs, datacenters)

Improving HTTP Performance:

Content Distribution Networks

- Caching and replication as a service
- Large-scale distributed storage infrastructure (usually) administrated by one entity
  - e.g., Akamai has servers in 20,000+ locations

- Combination of (pull) caching and (push) replication
  - **Pull:** Direct result of client requests
  - **Push:** Expectation of high access rate

- Also do some processing
  - Handle *dynamic* webpages
  - *Transcoding*

Improving HTTP Performance:

CDN Example - Akamai

- Akamai creates new domain names for each of their clients
  - e.g., *a128.g.akamai.net* for *cnn.com*

- The CDN's DNS servers are authoritative for the new domains

- The client content provider modifies its content so that embedded URLs reference the new domains
  - "Akamaize" content
  - e.g., *http://www.cnn.com/image-of-the-day.gif* becomes *http://a128.g.akamai.net/image-of-the-day.gif*

- Requests now sent to CDN's infrastructure…

Cost-effective Content Delivery

- **General theme:** multiple sites hosted on shared physical infrastructure
  - Efficiency of statistical multiplexing
  - Economies of scale (volume pricing, etc.)
  - Amortization of human operator costs

- Examples:
  - Web hosting companies
  - CDNs
  - Cloud infrastructure

Questions?

Taking Stock

Course so far:

- **Concepts, *Overall Architecture***
- **Network Layer,** *Best-effort global delivery of packets*
- **Transport Layer,** *Reliable (or unreliable) delivery of data*
- **Application Layer,** *DNS, HTTP*

Next:

- **Link Layer,** *Ethernet*

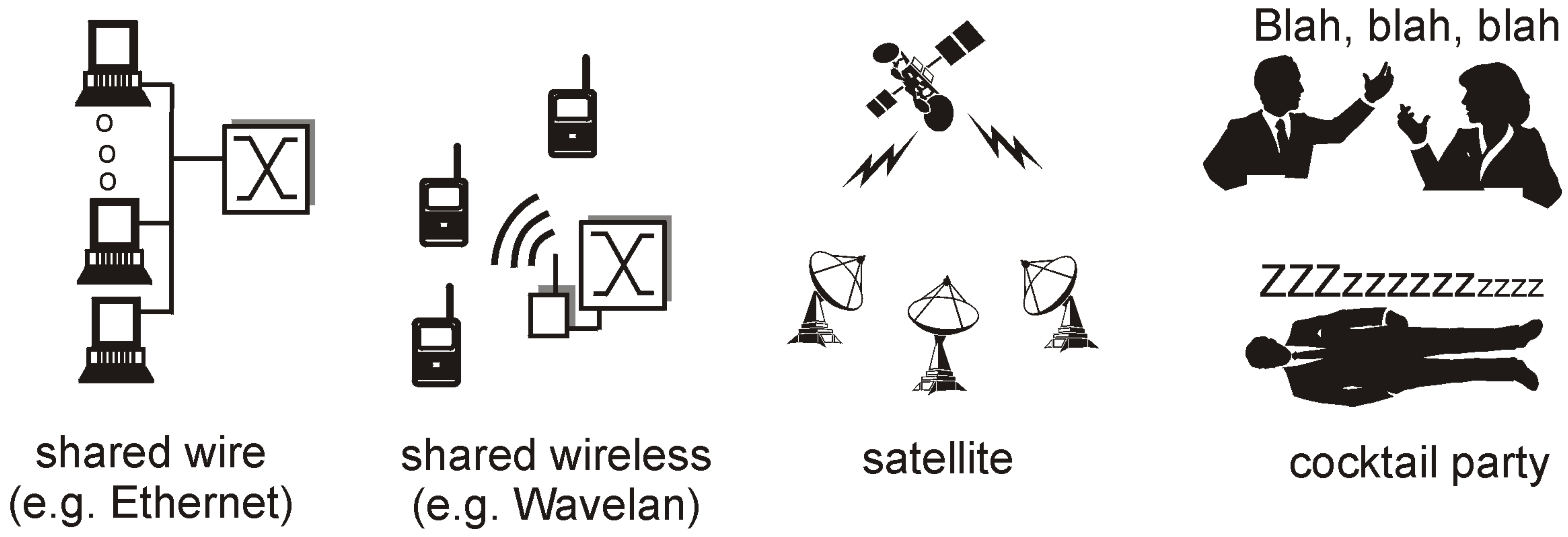
The Link Layer: Ethernet

CPSC 433/533, Spring 2021

Anurag Khandelwal

Point-to-Point vs. Broadcast Media

- **Point-to-point:** *dedicated* pairwise communication
    - E.g., long-distance fiber link
    - E.g., point-to-point link between Ethernet switch and host
- **Broadcast:** *shared* wire or medium
    - Traditional Ethernet (pre ~2000)
    - 802.11 Wireless LAN



Context: Shared Broadcast Channel

- Must avoid having multiple nodes speaking at once
- Otherwise, collisions can lead to garbled data
- Need distributed algorithm for sharing the channel
- Algorithm determines which node can transmit

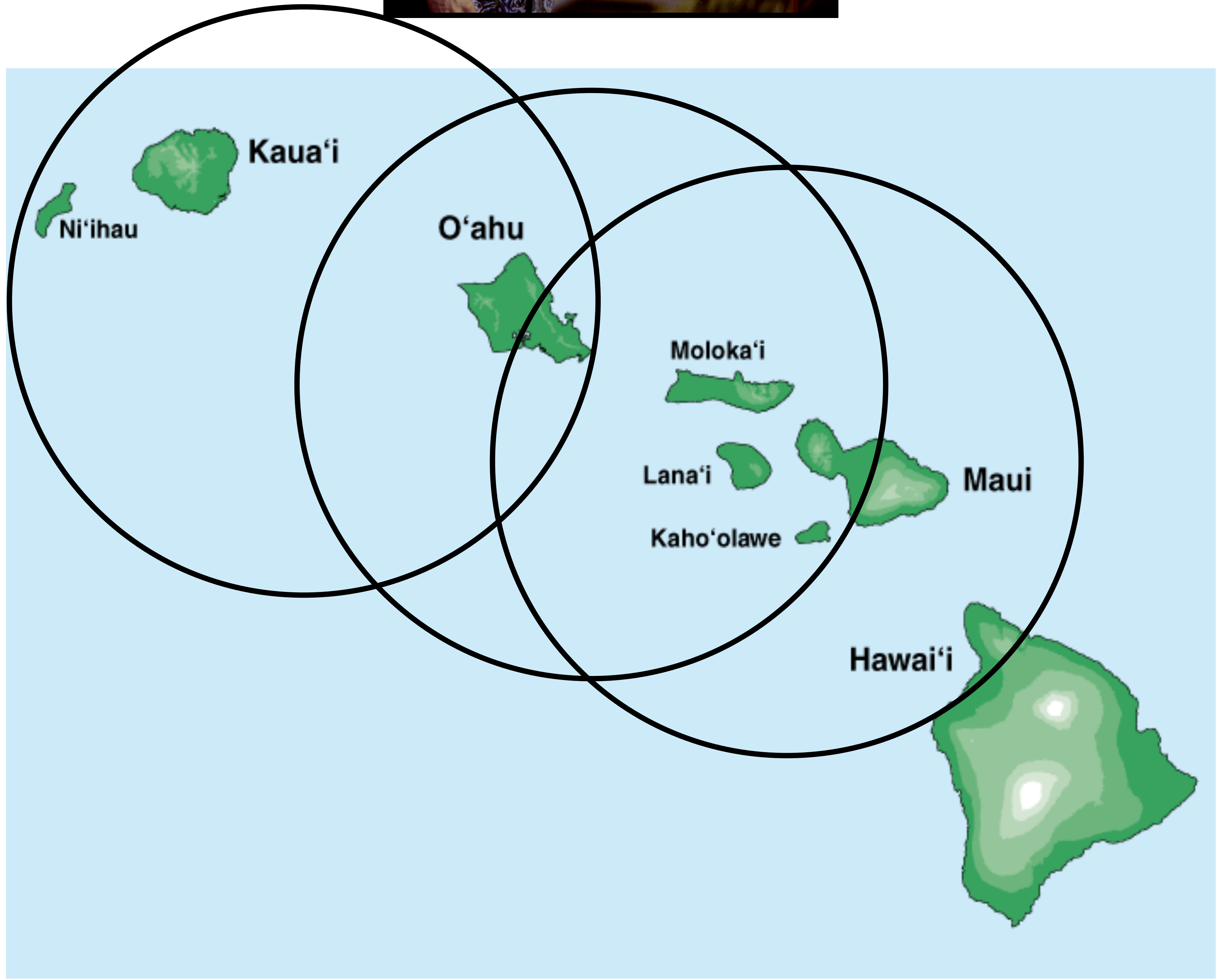
Multiple Access Algorithm

- *Channel partitioning:* divide channel by frequency (Frequency Division Multiplexing)
  - Can be wasteful! Only so much EM spectrum to go around, and many frequencies likely to be idle often (traffic is bursty)
- *Taking turns:* scheme for trading off who gets to transmit
  - Same drawbacks as FDM…
- *Random access:* allow collisions, and then recover
  - More in the Internet style!

Random Access MAC Protocols

- **When node has packet to send**
  - Transmit at full channel data rate
  - No a priori coordination among nodes
- **Two or more transmitted nodes → *collision***
  - Data lost
- **Random access MAC protocol specifies:**
  - How to detect collisions
  - How to recover from collisions
- **Examples**
  - ALOHA and Slotted ALOHA
  - CSMA, CSMA/CD, CSMA/CA (wireless, covered later)

Where it all started: AlohaNet



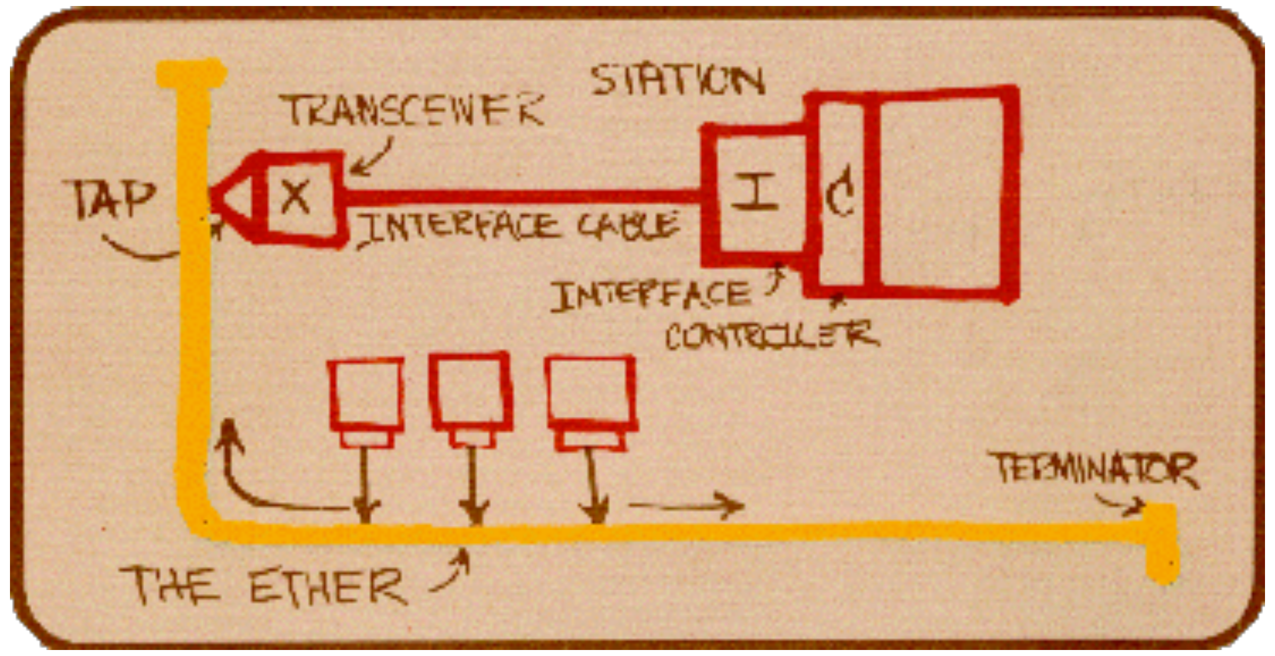
- Norm Abramson left Stanford in 1970 (*so he could surf!*)
- Setup the first data communication system for Hawaiian islands
- Central hub at U. Hawaii, Oahu

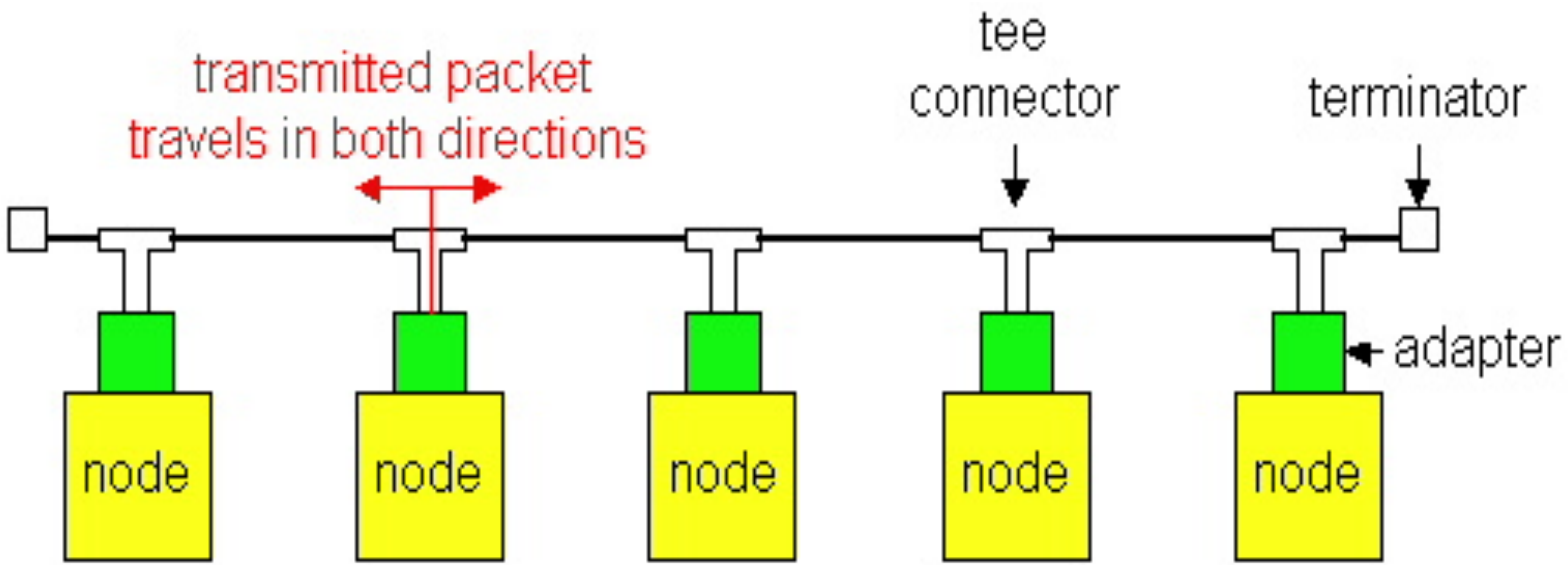
ALOHA Signaling

- Two channels: random access, broadcast
- Sites send packet to hub (random-access channel)
  - If not received (due to collision), site resends
- Hub sends packets to all sites (broadcast channel)
  - Sites can receive even if they are also sending

Ethernet

- **Bob Metcalfe, Xerox PARC, visits Hawaii and gets and idea!**
  - (Maybe we all need to go to Hawaii for good ideas…)
- **Shared wired medium**
  - coax cable





Evolution

- **Ethernet was invented as a broadcast technology**
  - Hosts share channel
  - Each packet received by all attached hosts
  - CSMA/CD for media access control

- **Current Ethernets are "switched" (later)**
  - Point-to-point links between switches; between a host and switch
  - No sharing, no CSMA/CD
  - Uses "self learning" and "spanning tree" algorithms for routing

Questions?

CSMA (Carrier Sense Multiple Access)

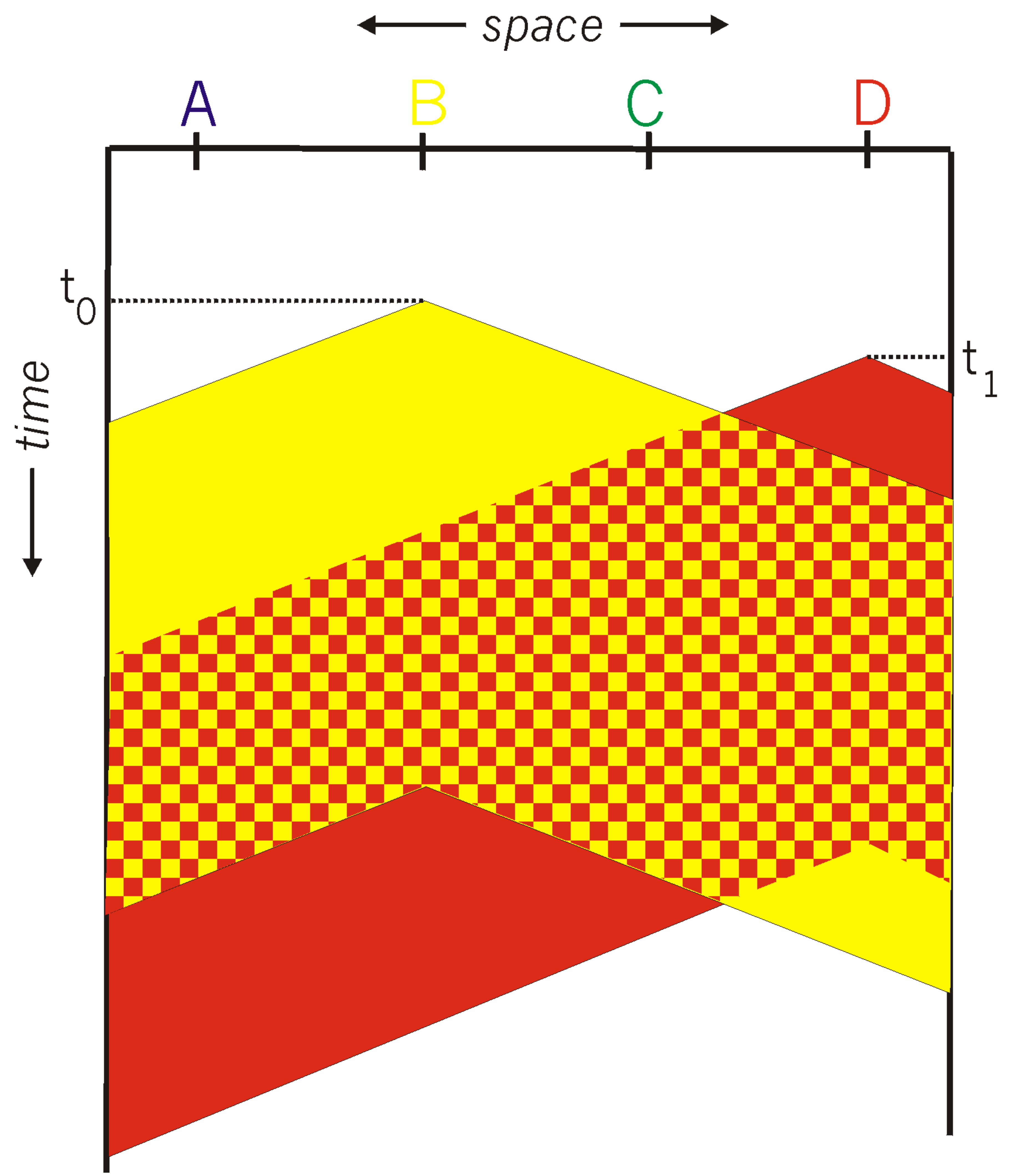
- CSMA: **Listen** before transmit
  - If channel sensed idle: transmit entire fame
  - If channel sensed busy: defer transmission

- Human analogy: don't interrupt others!

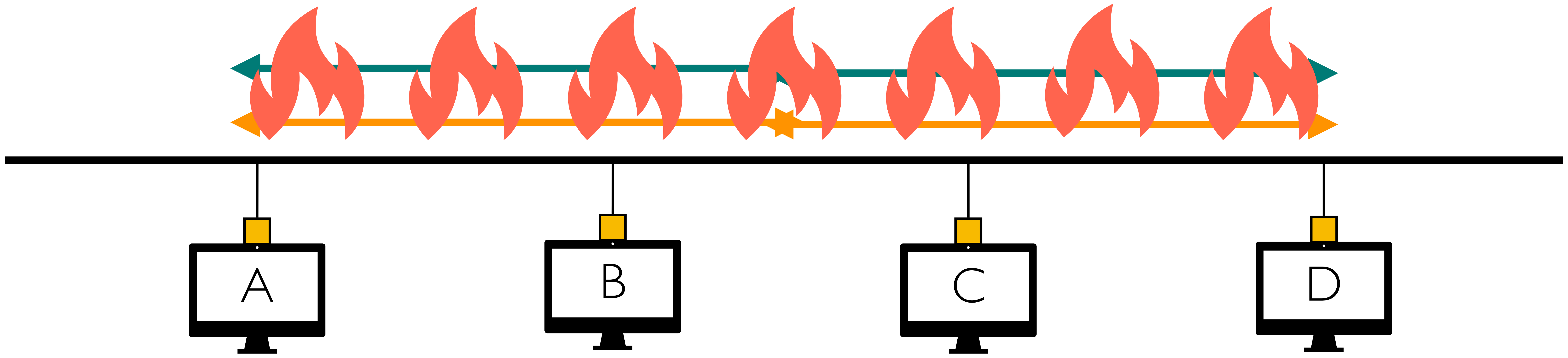
- Does this eliminate all collisions?
  - *No*, because of non-zero propagation delay

CSMA Collisions

- Propagation delay: two nodes may not each other before sending
- CSMA reduces but does not eliminate collisions
- Biggest remaining problem?
- Collisions still take full transmission slot!



CSMA Collisions

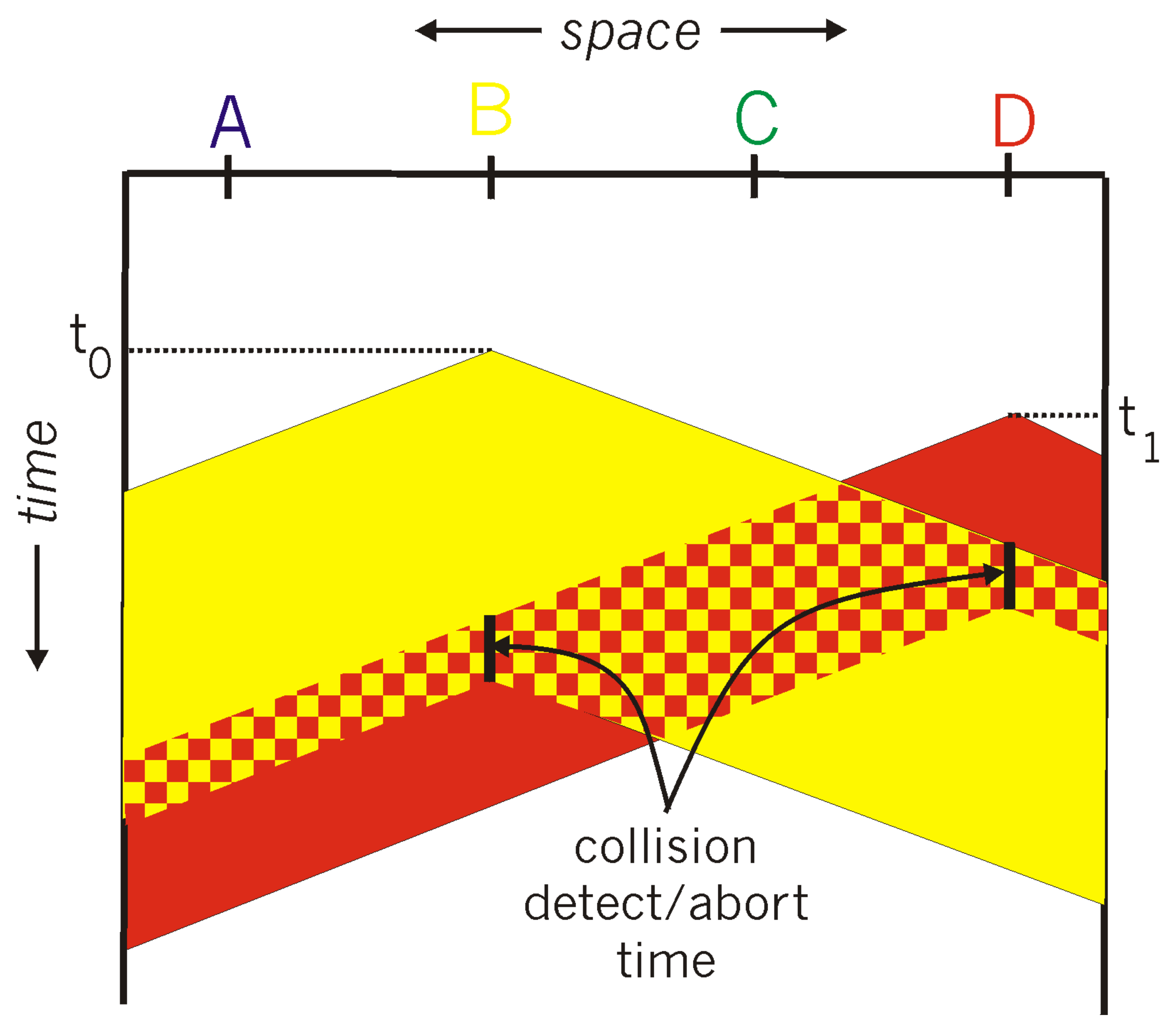


CSMA/CD (Collision Detection)

- **CSMA/CD: Carrier sensing, deferral as in CSMA**
  - *Collisions detected within short time*
  - Colliding transmissions aborted, reducing wastage
- **Collision detection easy in wired (broadcast) LANs**
  - Compare transmitted, received signals
- **Collision detection difficult in wireless LANs**
  - *Will learn more in Lecture on Wireless*

CSMA Collisions

- B and D can tell the collision occurred
- They will try again…
    - *When?*



How long should you wait?

- After collision, when should you resend?
- Should it be immediate?
- Should it be a random number with a fixed distribution?

Ethernet: CSMA/CD Protocol

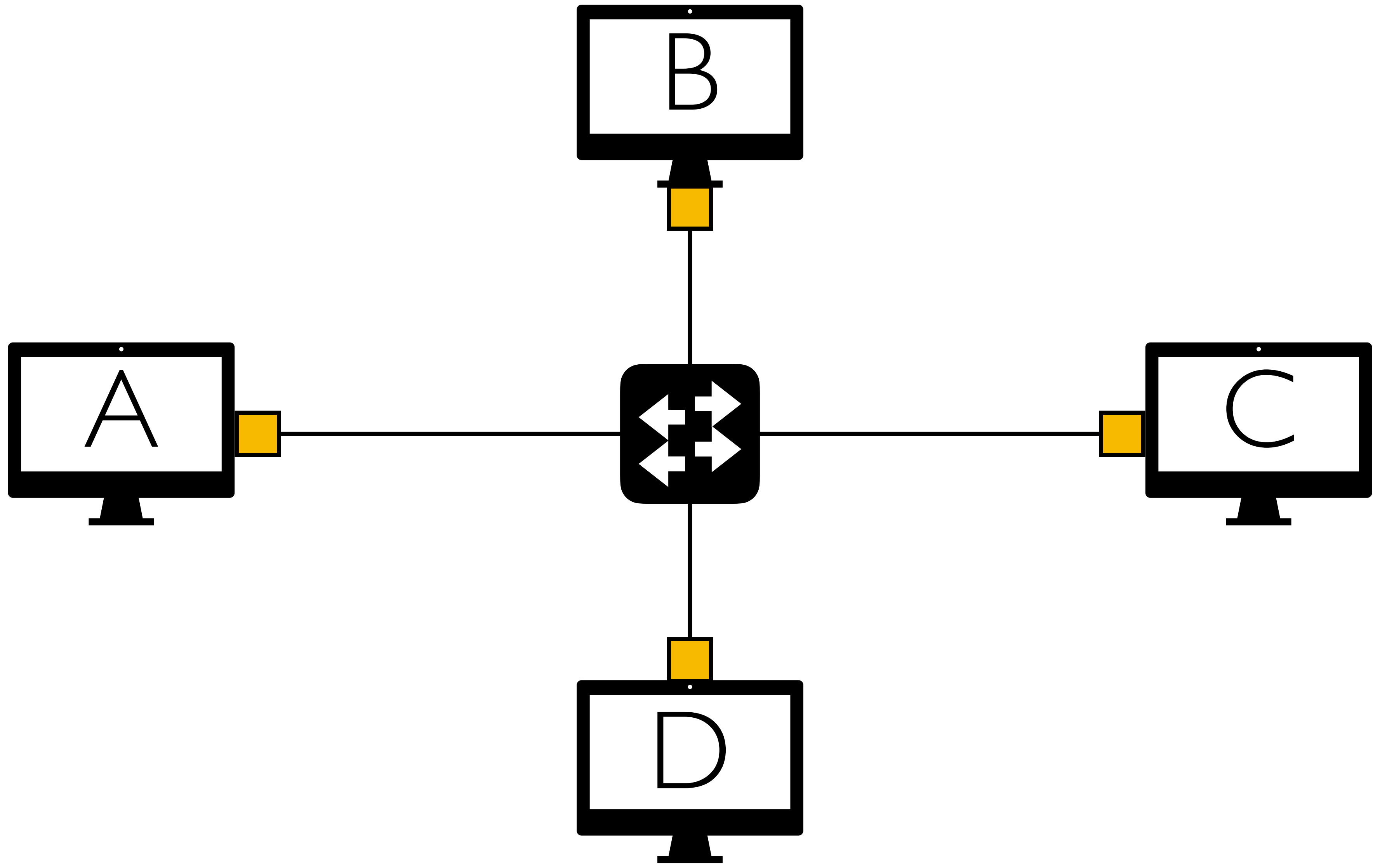
- **Carrier sense: wait for link to be idle**
- **Collision Detection: listen while transmitting**
  - No collision: transmission is complete
  - Collision: abort transmission & send *jam* signal
- **Random access: binary exponential backoff**
  - After collision, wait a random time before trying again
  - After m^{th} collision, chose K randomly from $\{0, \ldots, 2^{m-1}\}$
  - …and wait for K units of time before trying again
    - If transmission occurring when ready to send, wait until end of transmission (CSMA)

Questions?

Broadcast vs. Switched Ethernet

- **Ethernet was invented as a broadcast technology**
  - Each packet received by all attached hosts
  - CSMA/CD for media access control
- **Current Ethernets are "switched"**
  - Point-to-point links between switches and to hosts
  - No sharing, no CSMA/CD

Why Switched Ethernet?



- Enables concurrent communication
  - **Host A can talk to C, while B talks to D**
  - No collisions → no need for CSMA, CD
  - No constraints on link lengths, *etc.*

The evolution of Ethernet

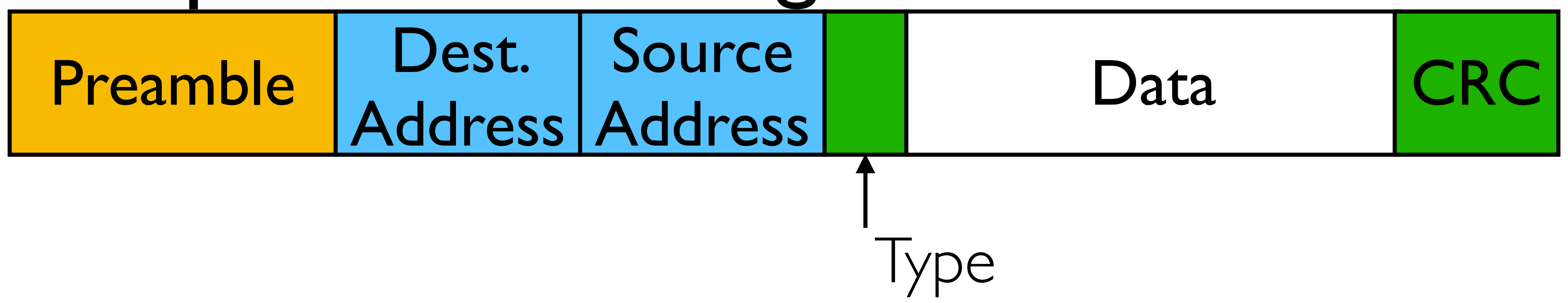
- From the shared media coax cables to dedicated links
- From 3 Mbit/s experimental ethernet to 100 Gbit/s
- From electrical signaling to optical
- **Changed almost everything except the frame format**
- Lesson: the **right interface** can accommodate many changes
  - Evolve the implementation while maintaining the interface (backward compatibility)

Topics

- Frames and framing
- Addressing
- Routing
- Forwarding
- Discovery: Bootstrapping end-to-end communication

Ethernet "Frames"

- **Encapsulates IP datagram**



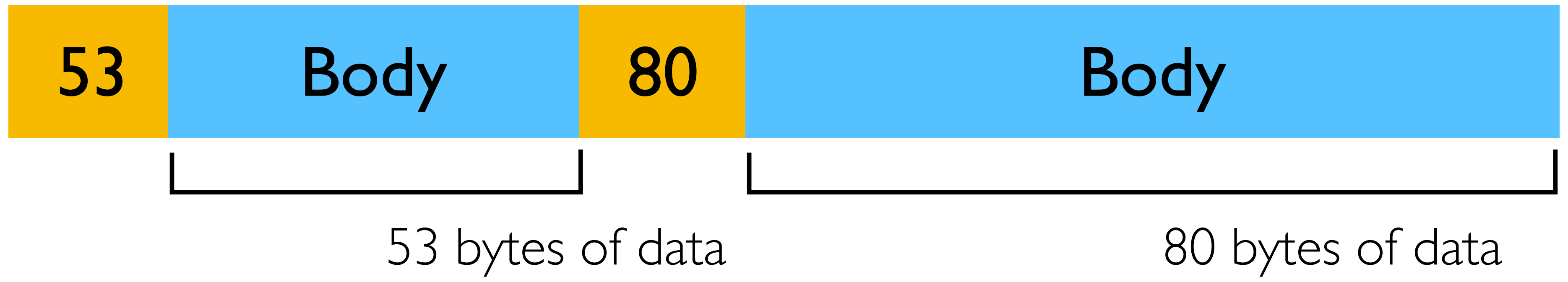
- **Preamble**
  - 7 bytes for clock synchronization
  - 1 byte to indicate start of frame
- **Addresses:** 6 bytes
- **Type:** 2 bytes, indicating higher-layer protocol (e.g., IP, Appletalk)
- **Data payload:** max 1500 bytes, minimum 46 bytes
- **CRC:** 4 bytes for error correction

Framing frames

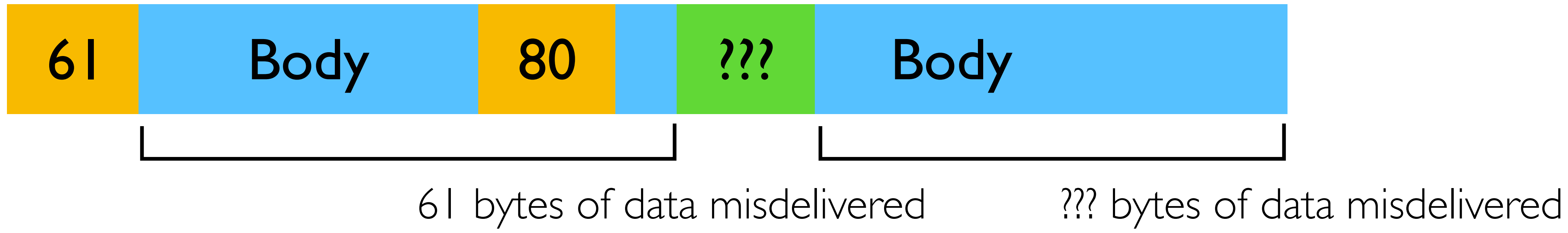
- Physical layer puts bits on a link
- But, two hosts connected on the same physical medium need to be able to exchange *frames*
  - Service provided by the link layer
  - Implemented by the network adapter
- Framing problem: how does the link layer determine where each frame begins and ends?

Simple approach: count bytes

- Sender includes number of bytes in header



- Receiver extracts this number of bytes of body
- But what if the count field is corrupted?



  - L2 will frame the wrong bytes → a framing error
  - CRC tells you to discard this frame, but what about the next one?

Desynchronization

- Once framing on a link is desynchronized, *it can stay that way*
- Need a method to *resynchronize*

Ethernet “Frames”

- Delineate frame with special “sentinel” bit pattern
  - E.g., 01111110 → start, 01111111 → end

01111110	Frame contents	01111111

- Problem: what if the sentinel bits occurs in the frame?
- **Solution:** bit stuffing
  - Sender always inserts a 0 after five 1s in the frame contents
  - Receiver always removes a 0 appearing after five 1s

Ethernet “Frames”

01111110	Frame contents	01111111

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

Example: Sentinel bits

- **Original data, including start/end of frame:**
  - 01111110011111101111101111100101111111

- **Sender rule: five 1s → insert a 0**
  - After bit stuffing at the sender:
  - 0111111001111101011111000111110001011111111

- **Receiver rule: five 1s and next bit 0 → remove 0**
  - After bit de-stuffing at the receiver:
  - 01111110011111101111101111100101111111

Questions?