CS 43: Computer Networks

24: Media Access-Link-Layer
Dec 5, 2019

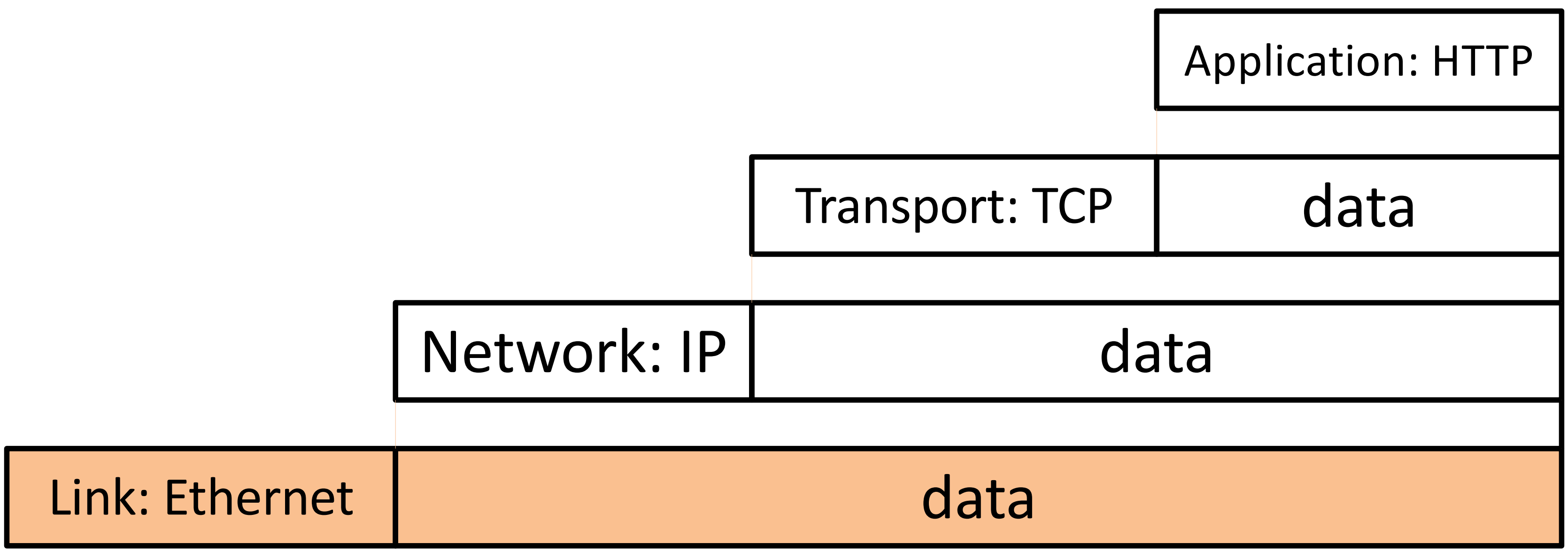
Adapted from Slides by: J.Kurose, J.Rexford, K. Webb

SWARTHMORE COLLEGE

Reading Quiz

Link Layer

- Function: Addressing, Framing, Media Access



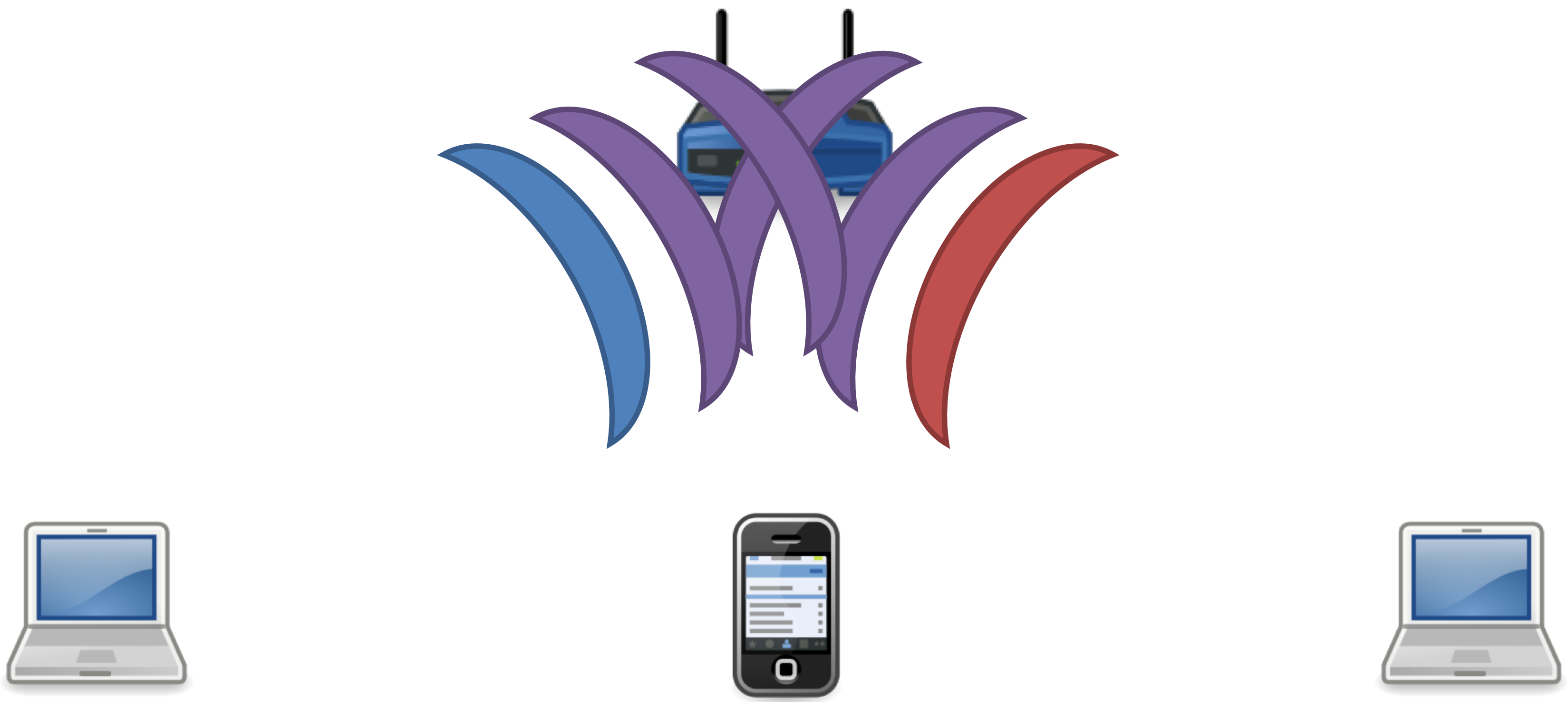
Last Class

- The link layer provides lots of functionality:
  - addressing, framing, media access, error checking
  - could be used independently of IP!
  - typically only small scale
- Many different technologies out there.
  - copper wires, optics, wireless, satellite
  - differing challenges for each

Link Access

- For wireless networks, this is a huge challenge.

Collision!



How should we handle collisions in general (for WiFi and other link media)?

A. Enforce at the end hosts that only one sender transmit at a time.

B. Enforce in the network that only one sender transmit at a time.

C. Detect collisions and retransmit later.

D. Something else.

Link Layer Functions

1. Addressing: identifying endpoints

2. Framing: Dividing data into pieces that are sized for the network to handle.

3. Link access: Determining how to share the medium, who gets to send, and for how long.

4. Error detection/correction and reliability.

Reliability in the link layer seems at odds with the E2E principle. Why would we add reliability here?

A. Legacy reasons: reliability was done at the link layer first, E2E came later.

B. It improves performance.

C. It’s necessary for correctness.

D. Some other reason.

E. It’s completely unnecessary.

Link Layer Functions

1. Addressing: identifying endpoints

2. Framing: Dividing data into pieces that are sized for the network to handle.

3. Link access: Determining how to share the medium, who gets to send, and for how long.

4. Error detection/correction and reliability.

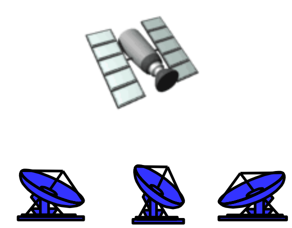
Multiple Access Links & Protocols

Two classes of "links":

- point-to-point
  - dial-up access
  - link between Ethernet switch, host
- *broadcast (shared wire or medium)*
  - old-fashioned Ethernet
  - 802.11 wireless LAN

shared wire (e.g., wired Ethernet)

shared RF (e.g., 802.11 WiFi)



shared RF (satellite)

Multiple Access Protocols

- Broadcast channel - every host hears every transmission
- If two or more nodes simultaneously transmit:
  - collision if node receives two or more signals at the same time

multiple access protocol

- algorithm that determines how nodes share channel, i.e., determine when node can transmit
- communication about channel sharing must use channel!
  - no out-of-band channel for coordination

An ideal multiple access protocol…

Given: broadcast channel of rate R bps

1. if only one node wants to transmit, it can send at rate R.
2. when M nodes want to transmit, each can send at average rate R/M (fairness)
3. fully decentralized:
   - no synchronization of clocks, slots
   - no special node to coordinate transmissions
4. simple

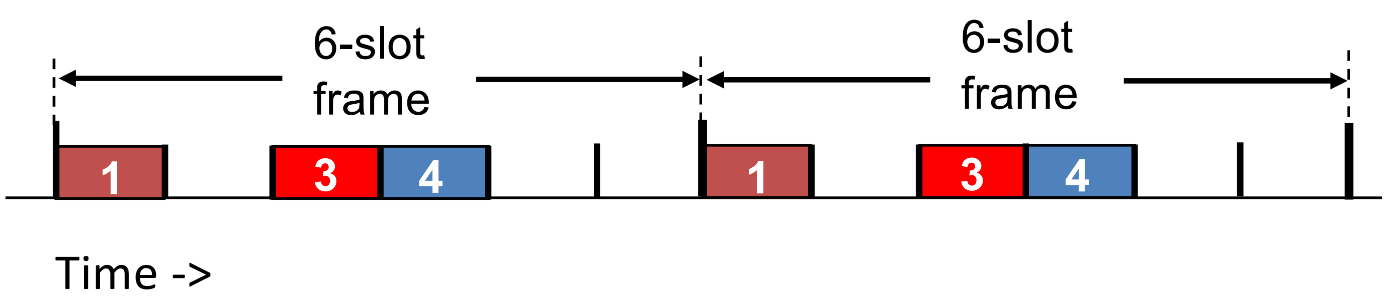
Media Access Control (MAC) Strategies

- channel partitioning
  - divide channel into smaller "pieces" (time slots, frequency, code)
  - allocate piece to node for exclusive use
- random access
  - channel not divided, allow collisions
  - "recover" from collisions
- taking turns
  - nodes coordinate with one another to take turns, share channel

Channel partitioning MAC protocols: TDMA

TDMA: time division multiple access

- Access to channel in "rounds", like round robin
- Each node gets fixed length time slot (length = pkt trans time) in each round
- Example: 6-station LAN, 1,3,4 have pkt, slots 2,5,6 idle

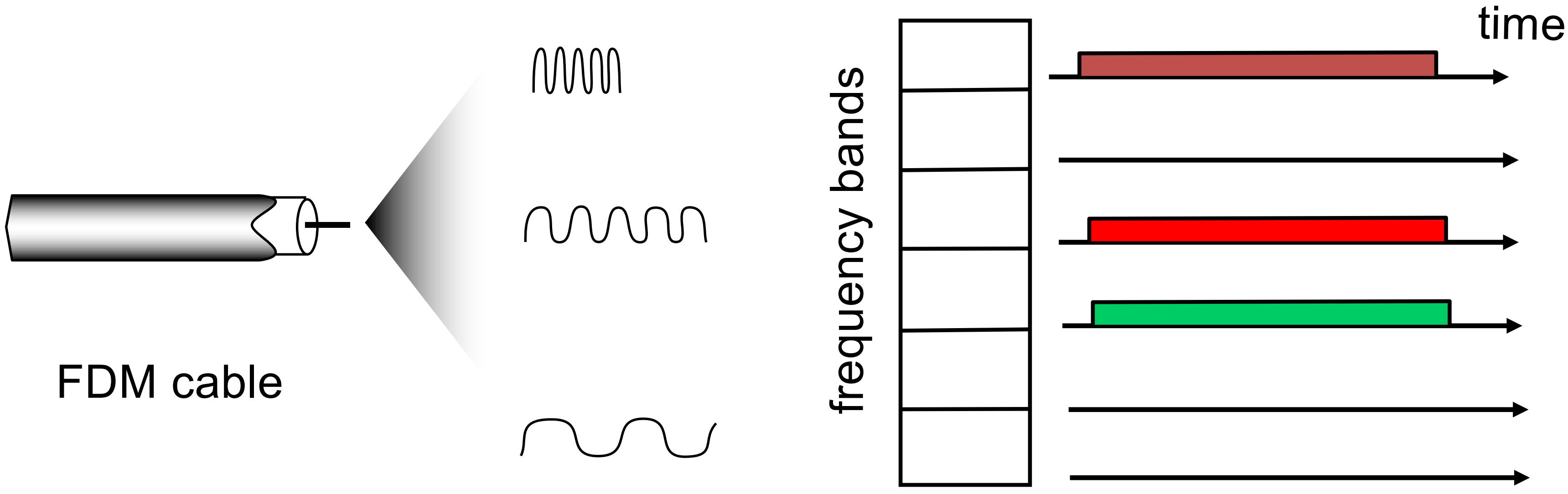


Channel partitioning MAC protocols: FDMA

FDMA: frequency division multiple access

- Channel spectrum divided into frequency bands
- Each node assigned a fixed frequency band

- Example: 6-station LAN, 1,3,4 have pkt, bands 2,5,6 idle



How many of our ideal properties does channel partitioning give us?

1. if only one node wants to transmit, it can send at rate R.
2. when M nodes want to transmit, each can send at average rate R/M (fairness)
3. fully decentralized:
   - no synchronization of clocks, slots
   - no special node to coordinate transmissions
4. simple

A. 0
B. 1
C. 2 (Which ones?)
D. 3
E. 4

Do we use channel partitioning?

- In what applications might this be a good idea?

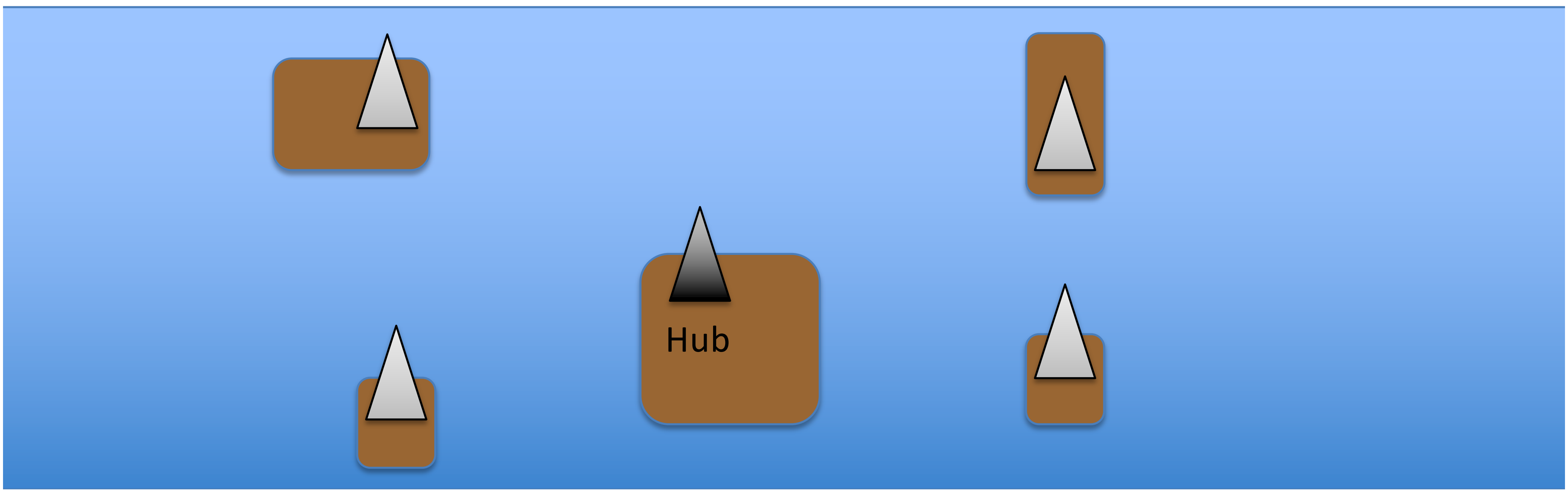
- Terrestrial radio/TV (frequency division)
- Satellite (frequency division)
- Fiber optic links (wavelength division)
- Cell phones
  - Old generations (time division)
  - Current generation (code division)

Random Access Protocols

- When node has a packet to send, try to send it
  - no a priori coordination among nodes
- Two or more transmitting nodes → "collision"
- random access MAC protocol specifies:
  - how to minimize collisions
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed retransmissions)

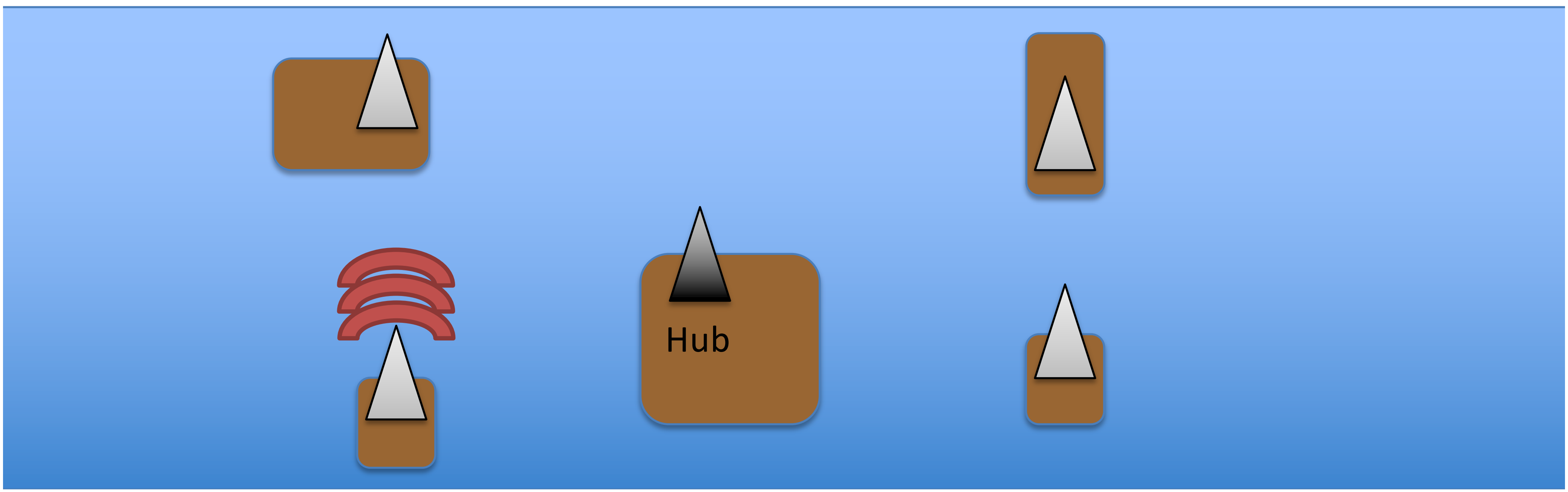
ALOHAnet (Unslotted / Pure)

- Norm Abramson at U of Hawaii in late 1960's
- Goal: network between islands
- Shared medium: radio



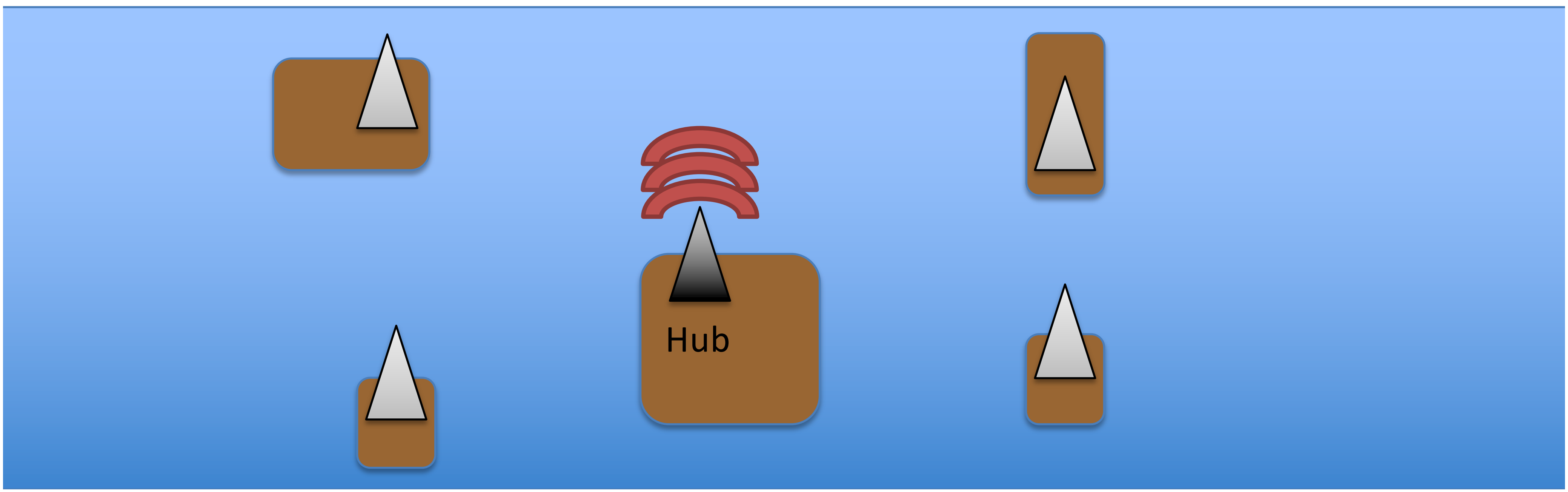
ALOHAnet

- Hub can hear everyone.
- If user gives you data, send it all, immediately.



ALOHAnet

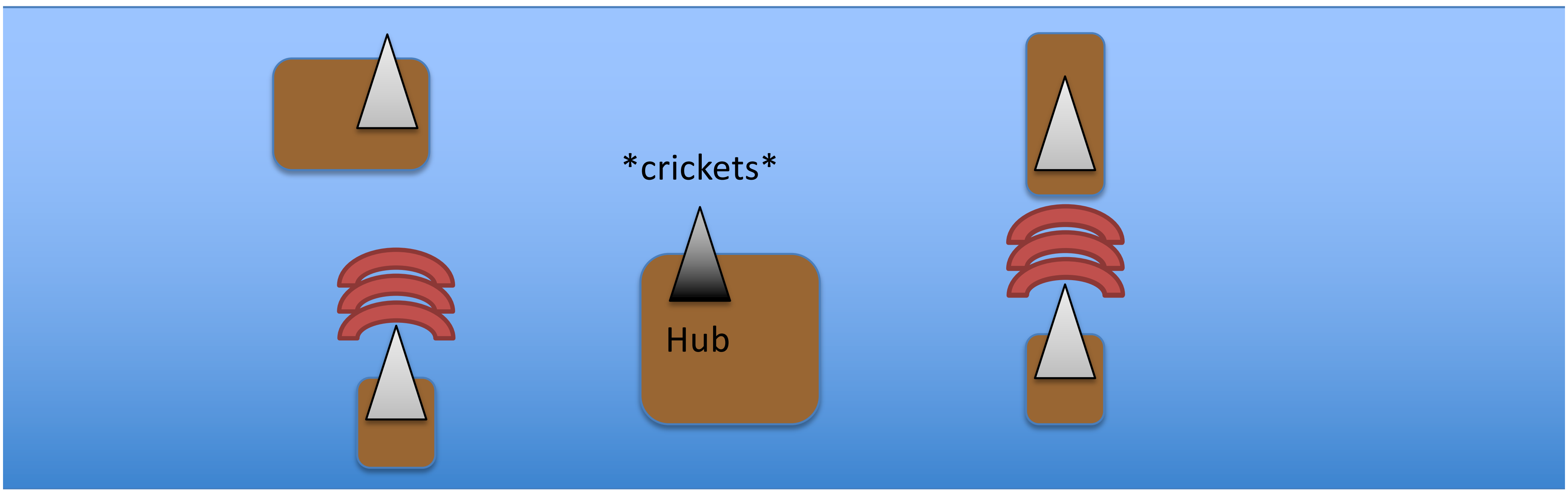
- If the hub received everything, it sends ACK.



ALOHAnet

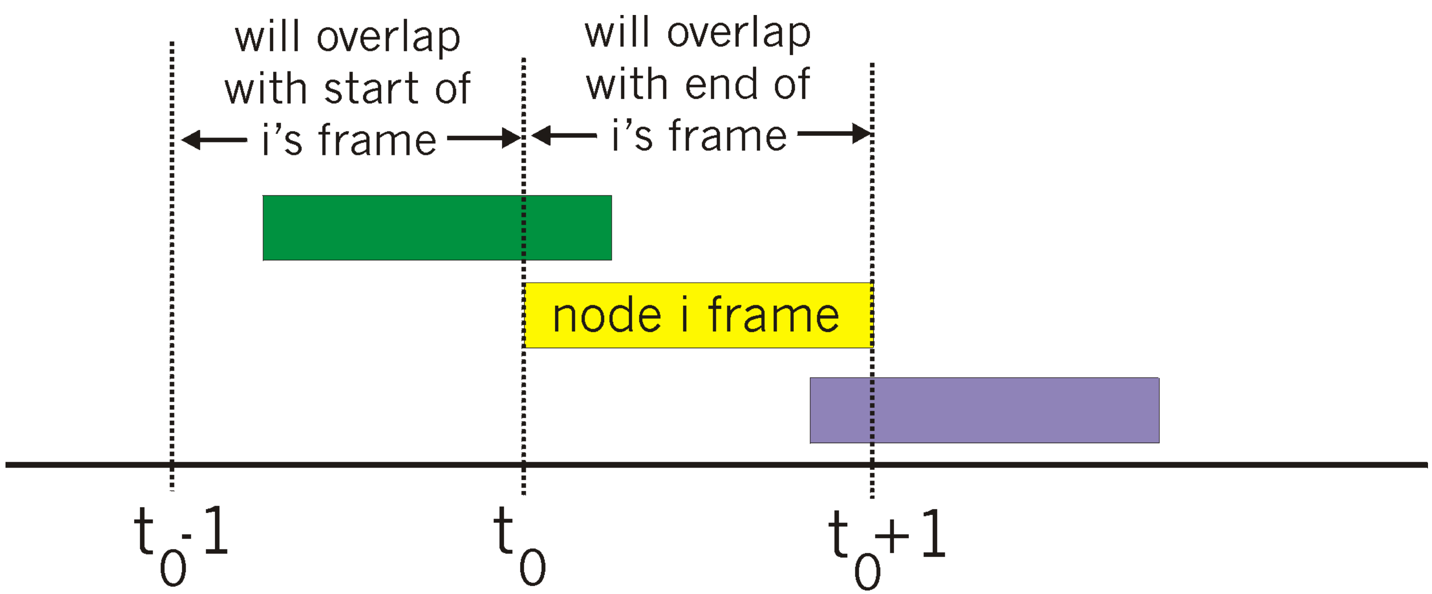
- If two senders collide…
- …hub sends back no ACKs.

- Senders wait a random time, send again.



(Unslotted / Pure) ALOHA

- Problems:
  - Sends immediately upon receiving data
  - Sends entire packets all at once



Carrier Sensing Multiple Access (CSMA)

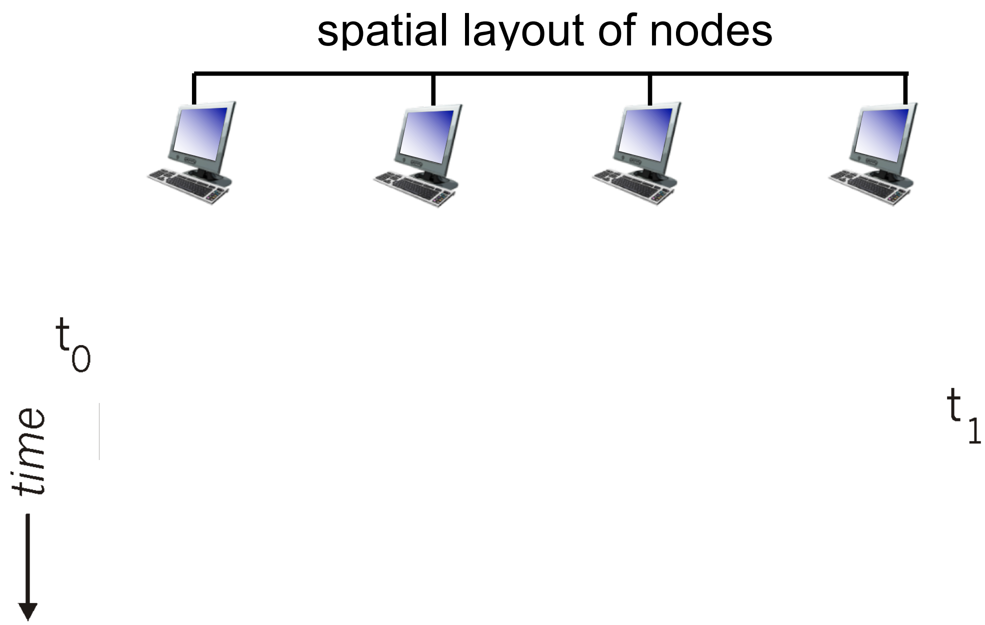
CSMA: listen before transmit:

if channel sensed idle: transmit

- if channel sensed busy, defer transmission
- human analogy: don't interrupt others!

CSMA collisions

- Collisions can still occur: propagation delay means two nodes may not hear each other's transmission
- Collision: entire packet transmission time wasted
  - distance & propagation delay play role in in determining collision probability



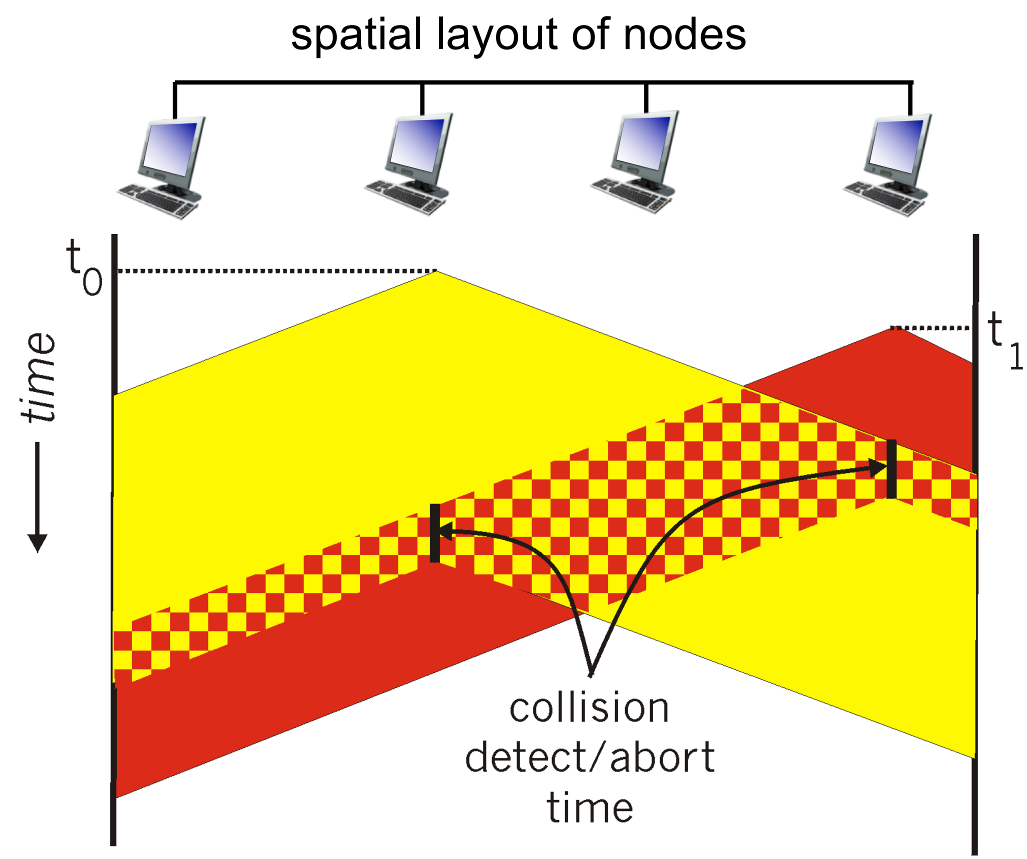
CSMA/CD (Collision Detection)

CSMA/CD: carrier sensing, deferral as in CSMA

- collisions detected within short time
- colliding transmissions aborted, freeing channel

- Collision detection:
  - easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - difficult in wireless LANs: received signal strength overwhelmed by local transmission strength

CSMA/CD (collision detection)



Ethernet and CSMA/CD

1. NIC (Network Interface Card) receives datagram from network layer, creates frame.
2. If NIC senses channel idle, starts frame transmission. If NIC senses channel busy, waits until channel idle, then transmits.
3. If NIC transmits entire frame without detecting another transmission, NIC is done with frame!
4. If NIC detects another transmission while transmitting, aborts and sends jam signal (maximize interference).
5. After aborting, NIC enters binary (exponential) backoff to send data again.

Exponential Back off

- After *m*th collision, NIC chooses *K* at random from *{0,1,2, …, 2^m-1}*.
- NIC waits K·512 bit times, then returns to checking if the channel is idle
- Longer back-off interval with more collisions

Like Human Conversation…

- Carrier sense
  - Listen before speaking
  - …and don't interrupt!
- Collision detection
  - Detect simultaneous talking
  - … and shut up!
- Random access
  - Wait for a random period of time
  - … before trying to talk again!

How many of our ideal properties does CSMA/CD give us?

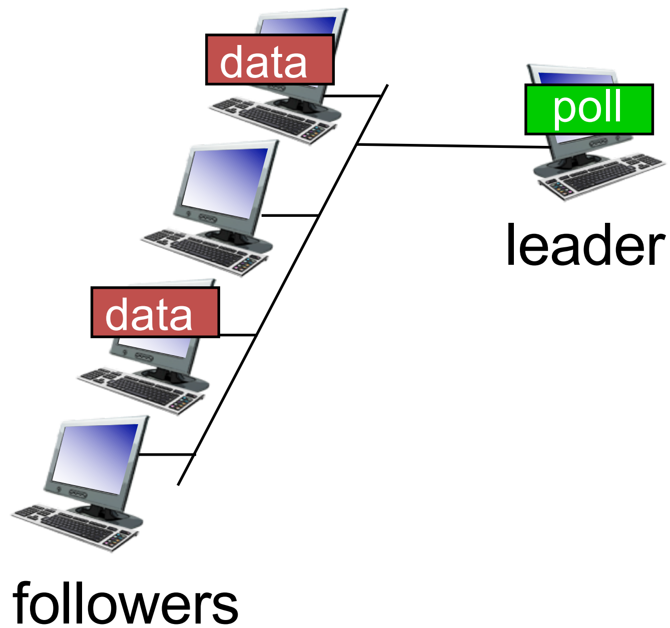
1. if only one node wants to transmit, it can send at rate R.
2. when M nodes want to transmit, each can send at average rate R/M (fairness)
3. fully decentralized:
   - no synchronization of clocks, slots
   - no special node to coordinate transmissions
4. simple

A. 0
B. 1
C. 2 (Which ones?)
D. 3
E. 4

"Taking turns" MAC protocols

Polling:

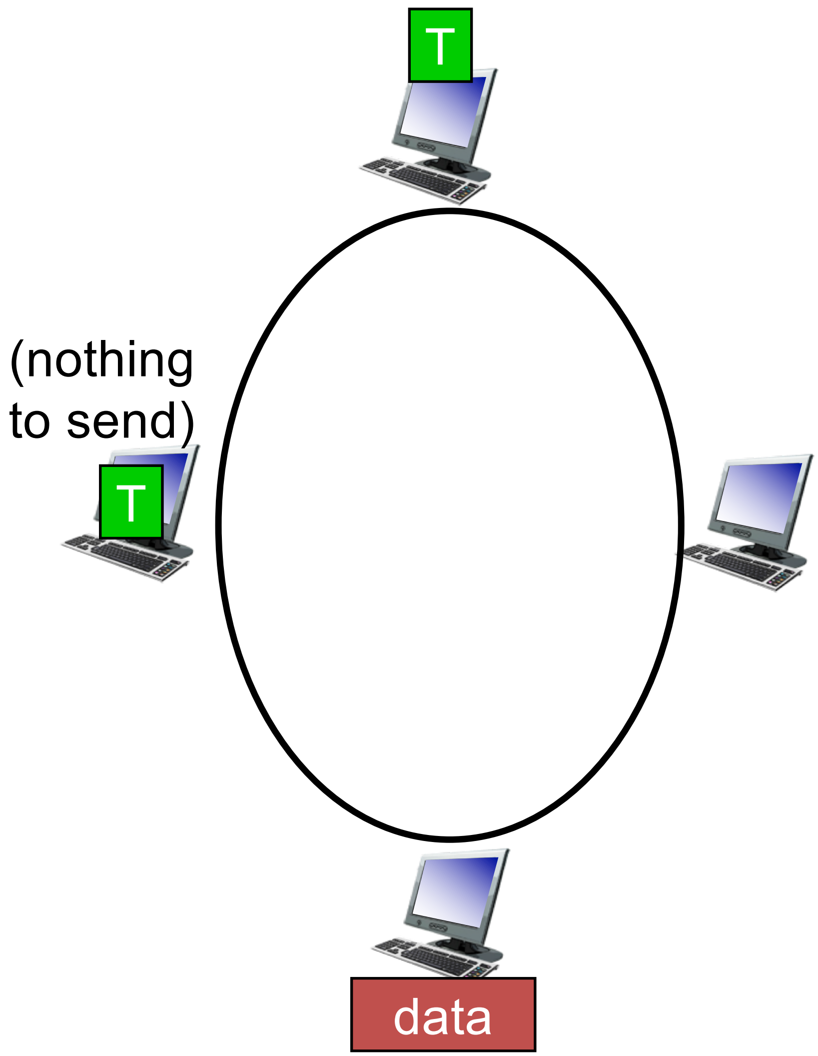
- leader node "invites" follower nodes to transmit in turn
- typically used with "dumb" follower devices
- Concerns:
  - polling overhead
  - latency
  - centralized leader



"Taking turns" MAC protocols

Token passing:

- Control token passed from one node to next sequentially.
- Can only transmit if holding the token.
- Limit on number of bytes sent per token.



How many of our ideal properties does taking turns (token passing) give us?

1. if only one node wants to transmit, it can send at rate R.
2. when M nodes want to transmit, each can send at average rate R/M (fairness)
3. fully decentralized:
   - no synchronization of clocks, slots
   - no special node to coordinate transmissions
4. simple

A. 0
B. 1
C. 2 (Which ones?)
D. 3
E. 4

In Practice…

- Techniques often combined.
  - Example: DOCSIS: Data Over Cable Service Interface Specifications: Cable Modem – Cable Provider
  - Frequency division of channels
  - TDMA Upstream with bandwidth contention (random access) requests by cable modems

- What about wireless Ethernet?
  - Old joke: "I don't know what the next link layer technology will look like, but I'm sure it will be named Ethernet."

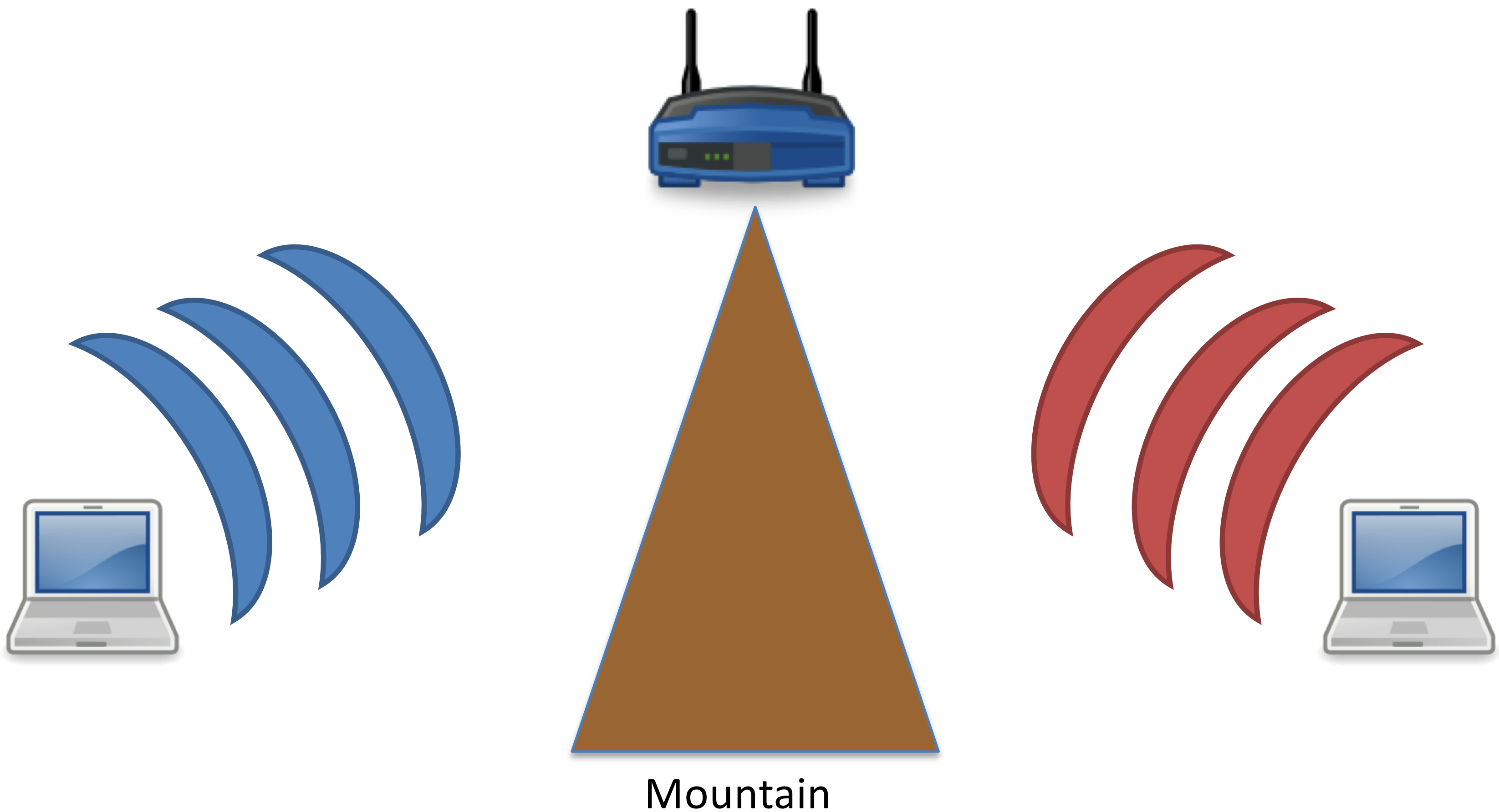
WiFi (802.11)

- Senders do carrier sensing like Ethernet.

Assumes hosts can hear each other!

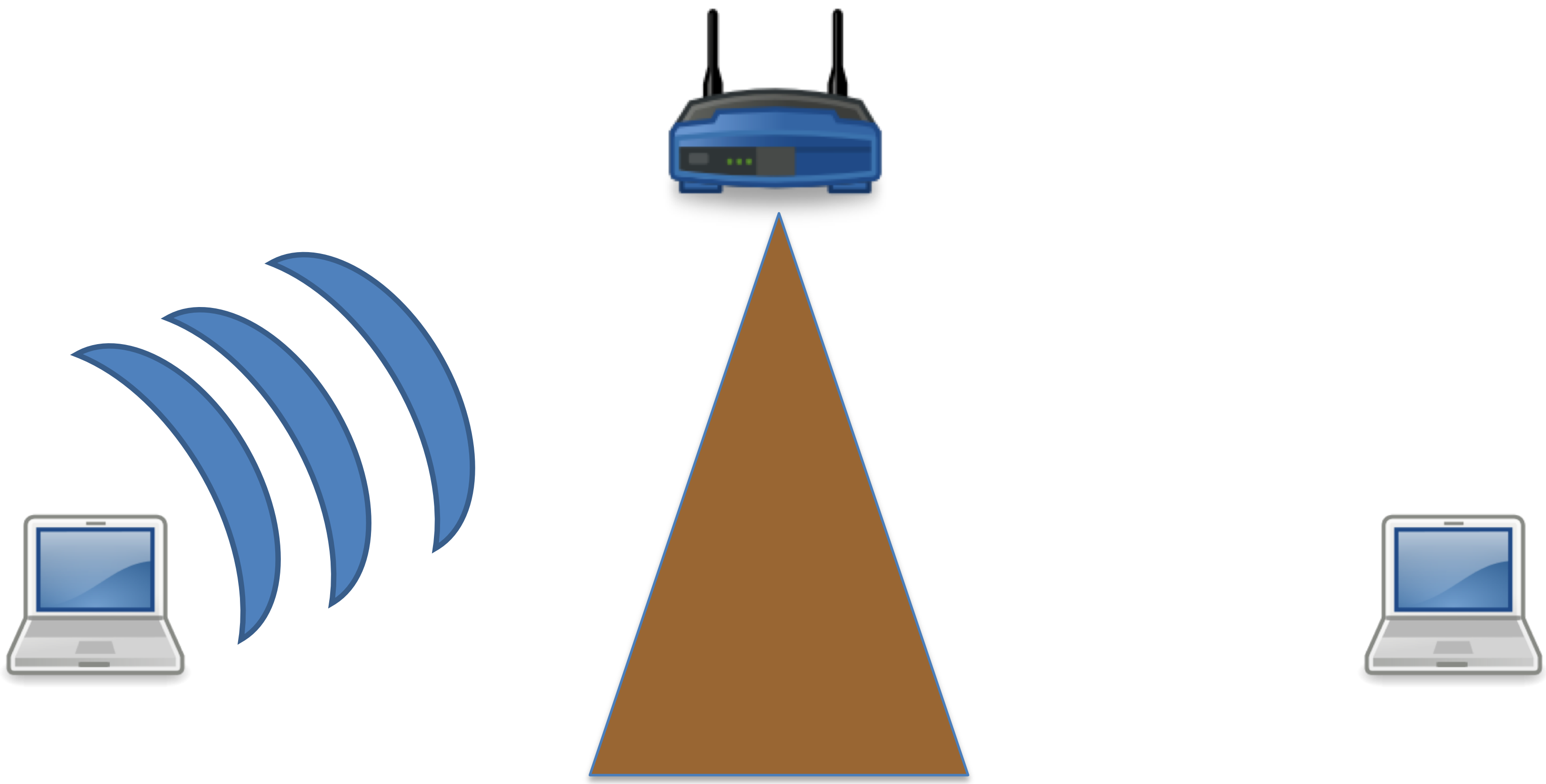
“Hidden Terminal” Problem

- Senders collide at receiver, but they can’t hear each other!



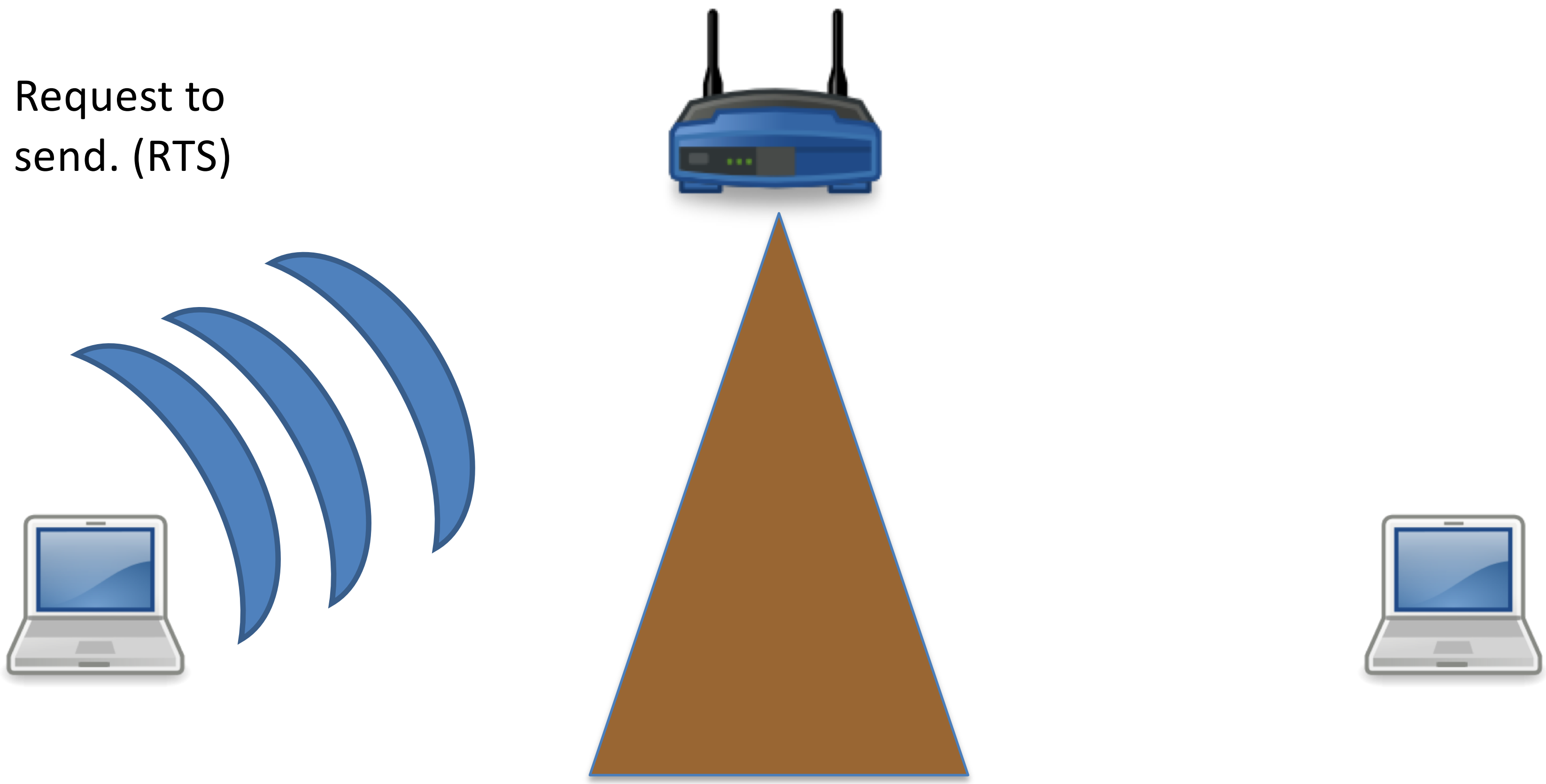
CSMA/CA (Collision Avoidance)

- If sending small (threshold configurable) frame, just send it.



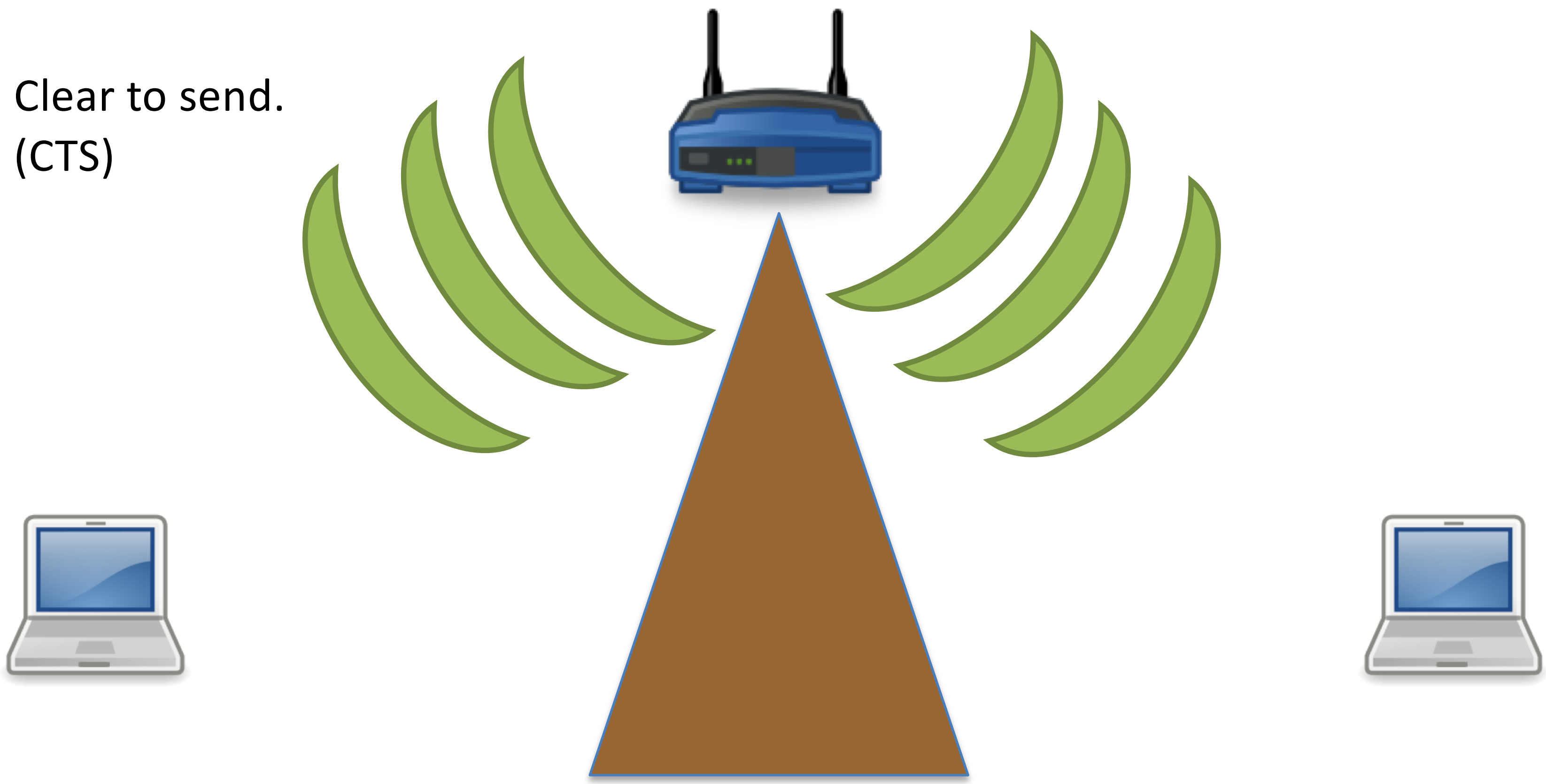
CSMA/CA (Collision Avoidance)

- If sending large frame, ask for permission first.



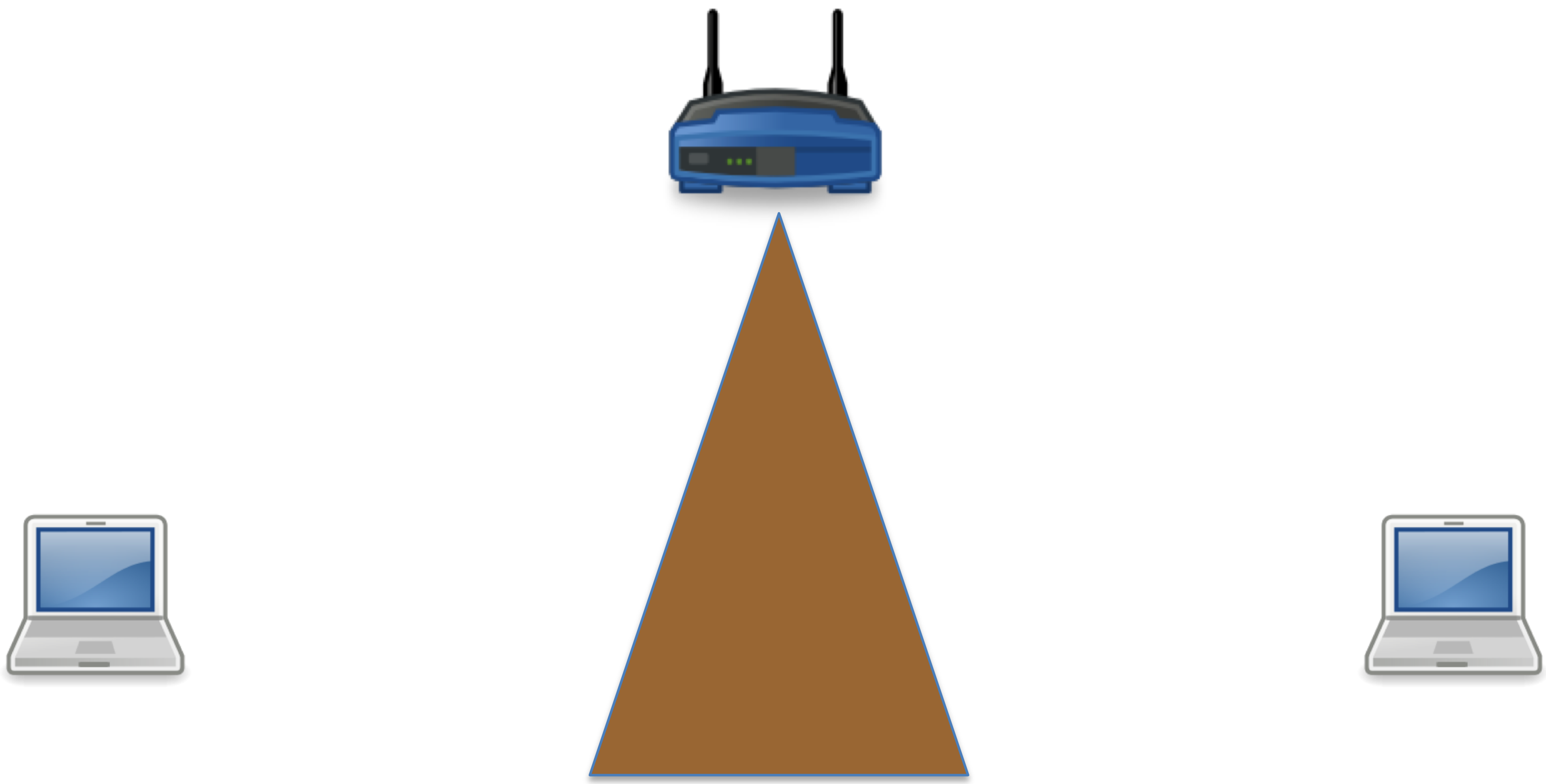
CSMA/CA (Collision Avoidance)

- If granted, it will be heard by everyone.



CSMA/CA (Collision Avoidance)

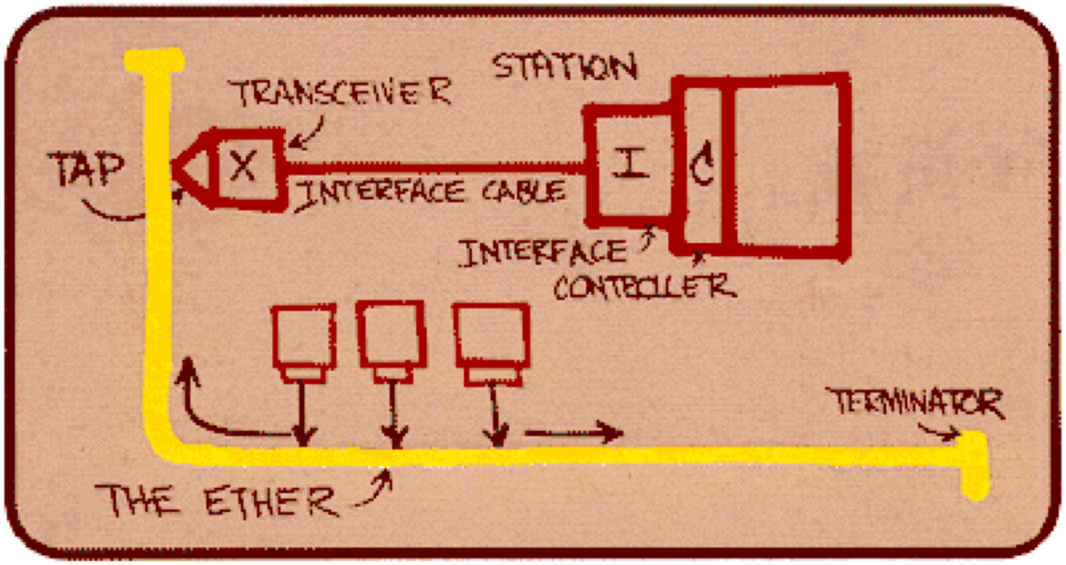
- RTS/CTS is like taking turns.



Summary of MAC protocols

- channel partitioning, by time, frequency or code
  - Time Division, Frequency Division
- random access (dynamic),
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - carrier sensing:
    - easy in some technologies (wire), hard in others (wireless)
  - CSMA/CD used in Ethernet
  - CSMA/CA used in 802.11
- taking turns
  - Polling from central site, token passing
  - Bluetooth, FDDI, token ring

Ethernet



Metcalfe's Ethernet sketch

"Dominant" wired LAN technology:

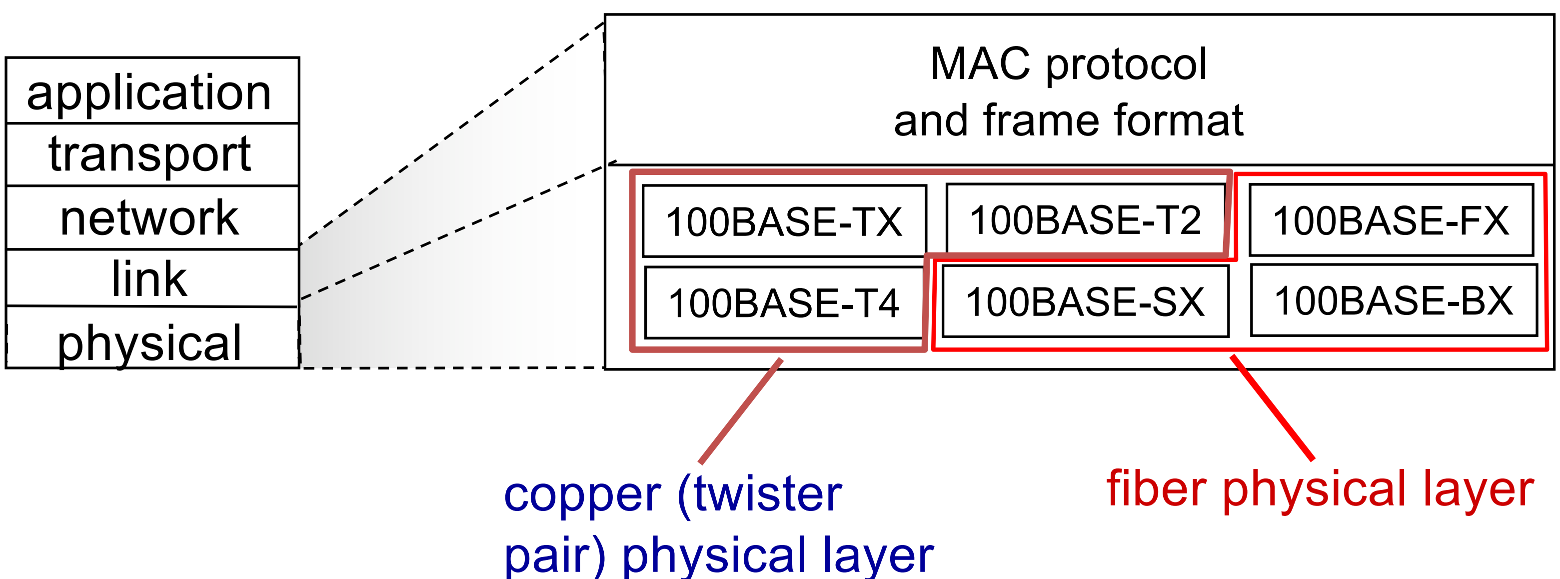
- cheap $20 for NIC
- first widely used LAN technology
- simpler, cheaper than token LANs and ATM
- kept up with speed race: 10 Mbps – 10 Gbps

Ethernet: unreliable, connectionless

- Connectionless: no handshaking between sending and receiving NICs
- Unreliable: receiving NIC doesn't send acks or nacks to sending NIC
  - data in dropped frames recovered only if initial sender uses higher layer reliable delivery (e.g., TCP), otherwise dropped data lost
- Ethernet's MAC protocol: CSMA/CD with binary exponential backoff

802.3 Ethernet standards: link & physical layers

- Many different Ethernet standards
  - Common MAC protocol and frame format
  - Speeds: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10Gbps
  - Physical layer media: fiber, copper cable



Ethernet frame structure

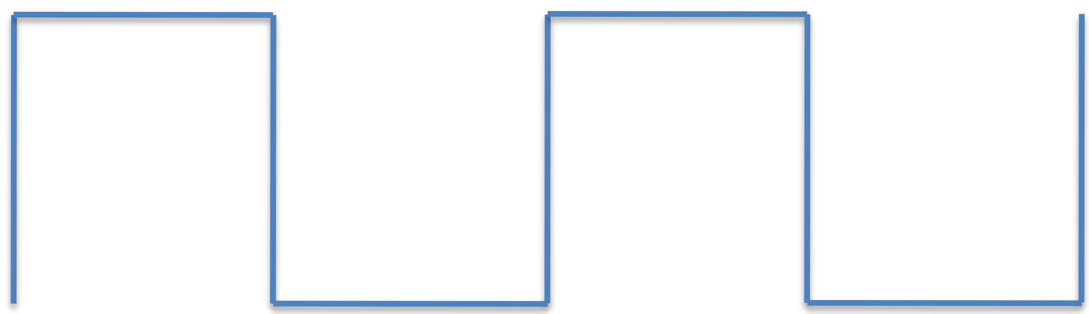
Sender encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame

preamble:

- 7 bytes with pattern 10101010 followed by one byte with pattern 10101011

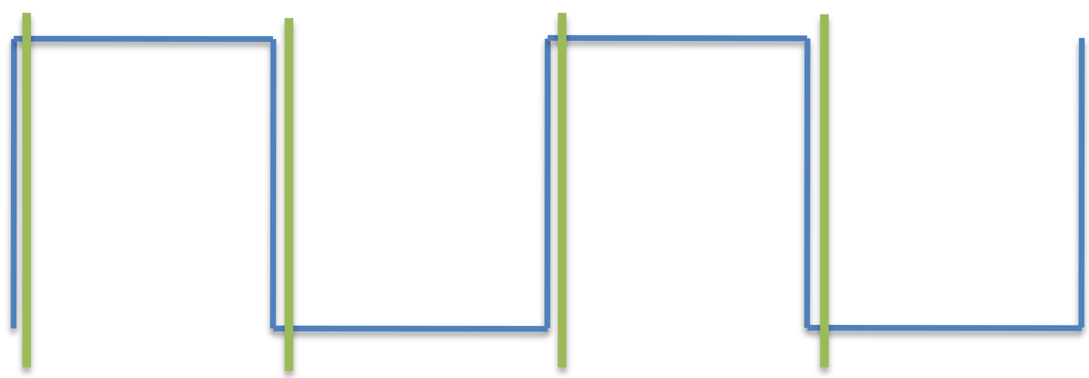
Clock Synching

- Bits represented as voltages, either low or high
- We will read one bit per clock cycle



Clock Synching

- Bits represented as voltages, either low or high
- We will read one bit per clock cycle

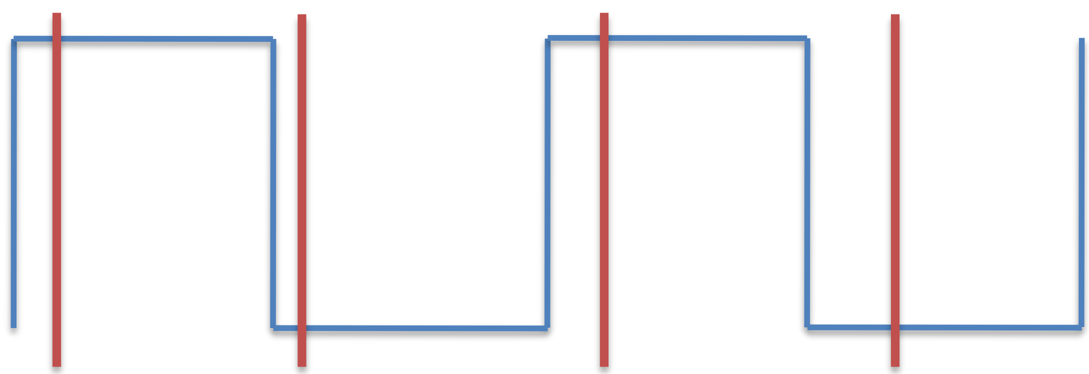


Ideal receiver: Sample signal at regular interval.

For 1 Gbps Ethernet, ~1 nanosecond interval.

Clock Synching

- Bits represented as voltages, either low or high
- We will read one bit per clock cycle

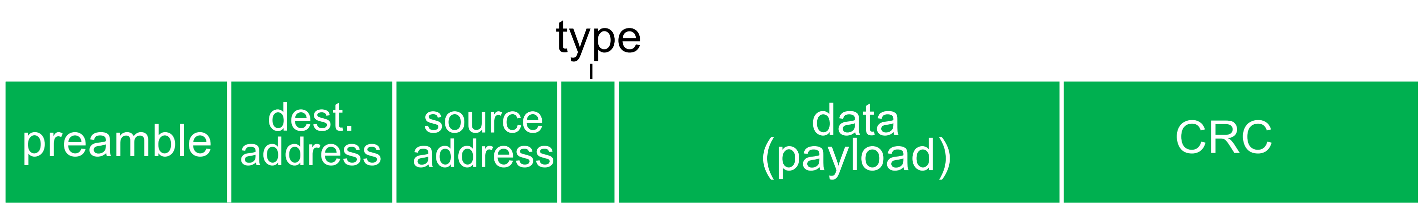


Problem: receiver clock may not agree with sender!

Preamble let's receiver see several 0 -> 1 -> 0 -> ... transitions.

Ethernet frame structure (more)

- addresses: 6 byte source, destination MAC addresses
  - if adapter receives frame with matching destination address, or with broadcast address (e.g. ARP packet), it passes data in frame to network layer protocol
  - otherwise, adapter discards frame
- type: indicates higher layer protocol (mostly IP but others possible, e.g., Novell IPX, AppleTalk)
- CRC: cyclic redundancy check at receiver
  - error detected: frame is dropped



MAC Addresses

- MAC (or LAN or physical or Ethernet) address:
  - 48 bit MAC address
  - e.g.: 1A-2F-BB-76-09-AD

hexadecimal (base 16) notation
(each digit represents 4 bits)

MAC vs. IP Addresses

- 32-bit IP address
- IP hierarchical address not portable!
  - address depends on IP subnet to which node is attached
- used by network layer for end-to-end routing

- 48 bit MAC address burned in NIC ROM.
- MAC flat address: portability
  - can move LAN card from one LAN to another
- used locally to get from one interface to another physically-connected interface

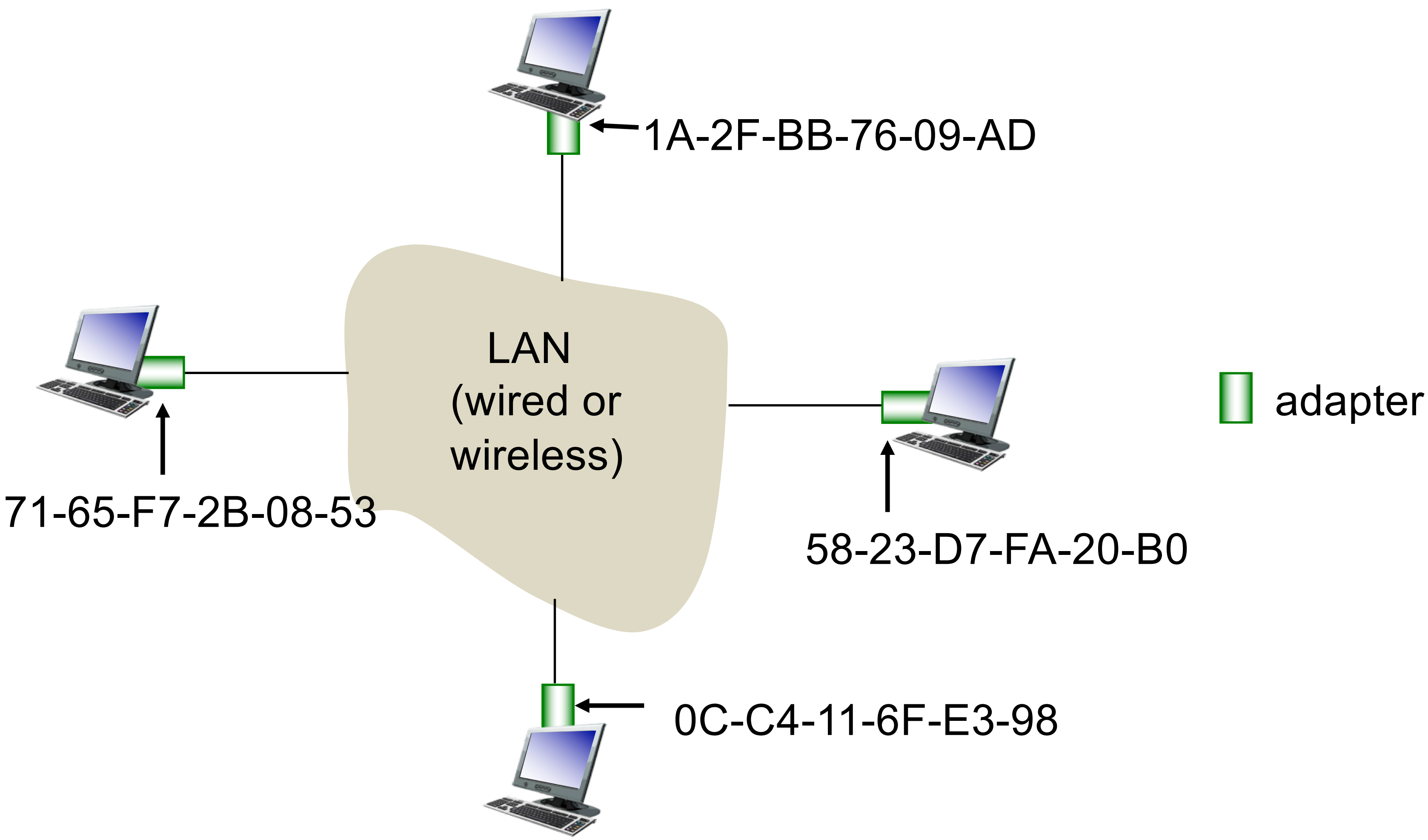
Analogy:

MAC address: like Social Security Number

IP address: like postal address

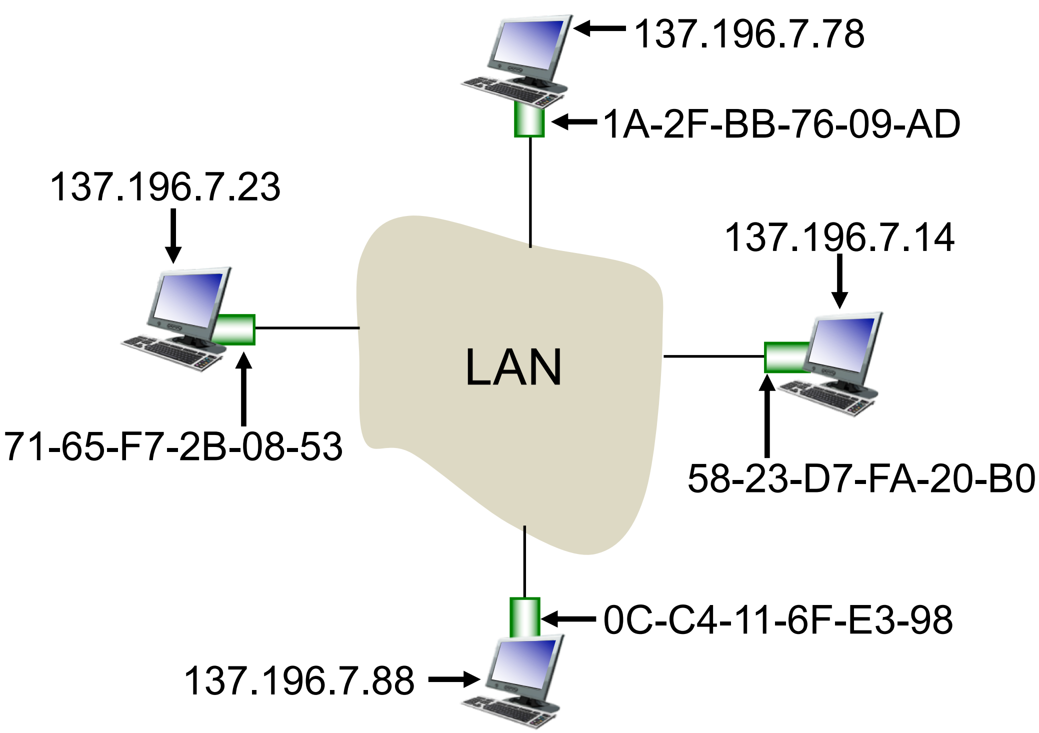
MAC Addresses

Each interface/adapter on LAN has unique MAC address



ARP: Address Resolution Protocol

Question: how to determine interface's MAC address, knowing its IP address?



ARP table: each IP node (host, router) on LAN has table

- IP/MAC address mappings for some LAN nodes:

  < IP address; MAC address; TTL>

- TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

ARP protocol & LAN communication

- A wants to send datagram to B. A knows B’s IP address.
  - B’s MAC address not in A’s ARP table.

- A broadcasts ARP query packet, containing B's IP address
  - dest Ethernet address = FF-FF-FF-FF-FF-FF
  - all nodes on LAN receive ARP query, most ignore it

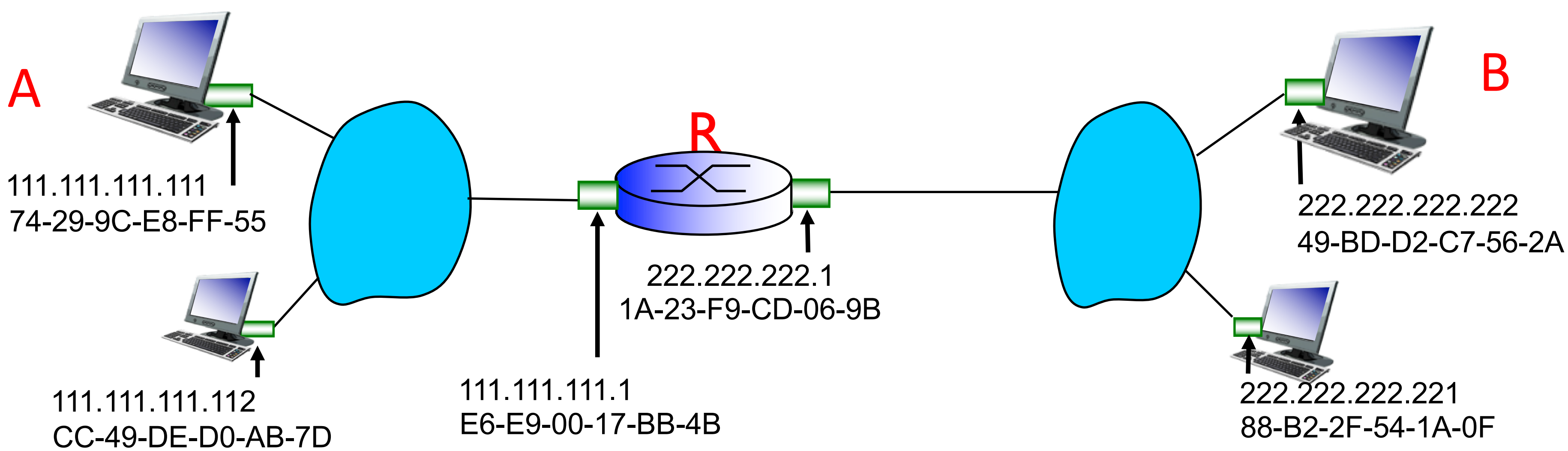
- B receives ARP packet, replies to A with its (B's) MAC address
  - frame sent to A’s MAC address (unicast)

- A caches IP-to-MAC address pair in its ARP table until timeout
  - soft state: times out unless refreshed, can be reacquired

Addressing: routing to another LAN

Walkthrough: send datagram from A to B via R

- focus on addressing – at IP and MAC layer
- assume A knows B's IP address (e.g., DNS lookup)
- how many subnets are present in this figure?

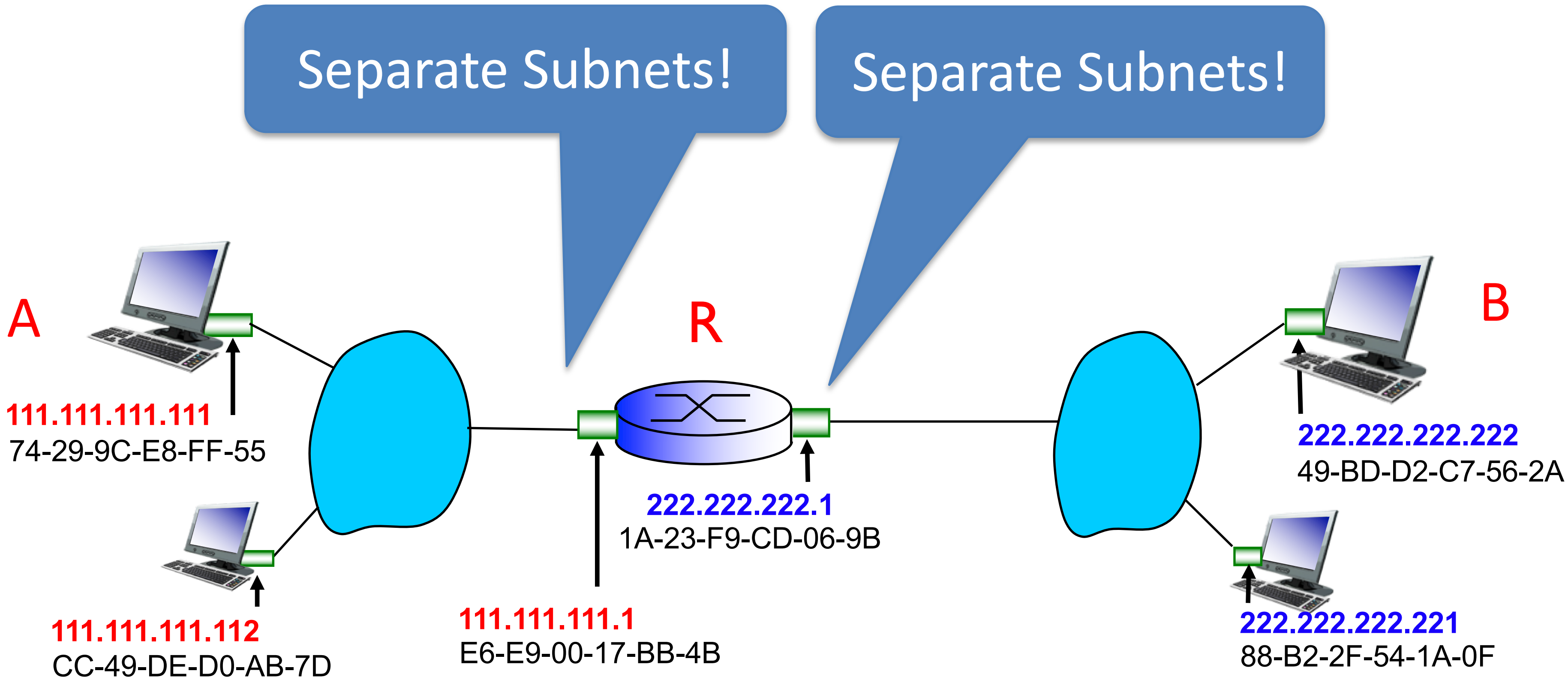


Addressing: routing to another LAN

Walkthrough: send datagram from A to B via R

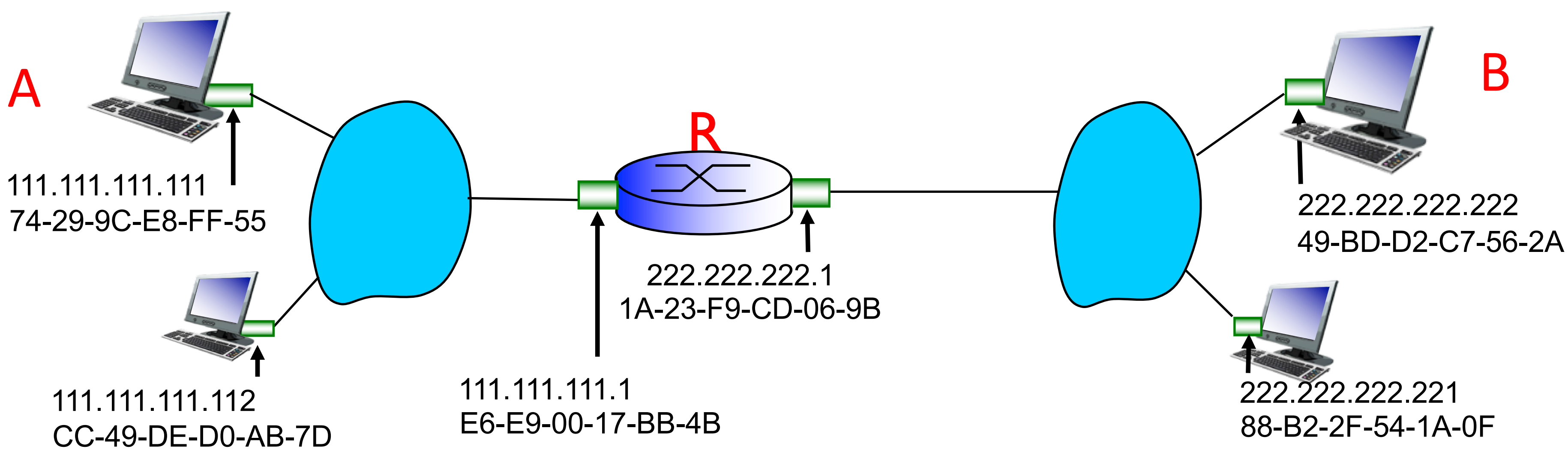
– focus on addressing – at IP and MAC layer

– assume A knows B's IP address (e.g., DNS lookup)



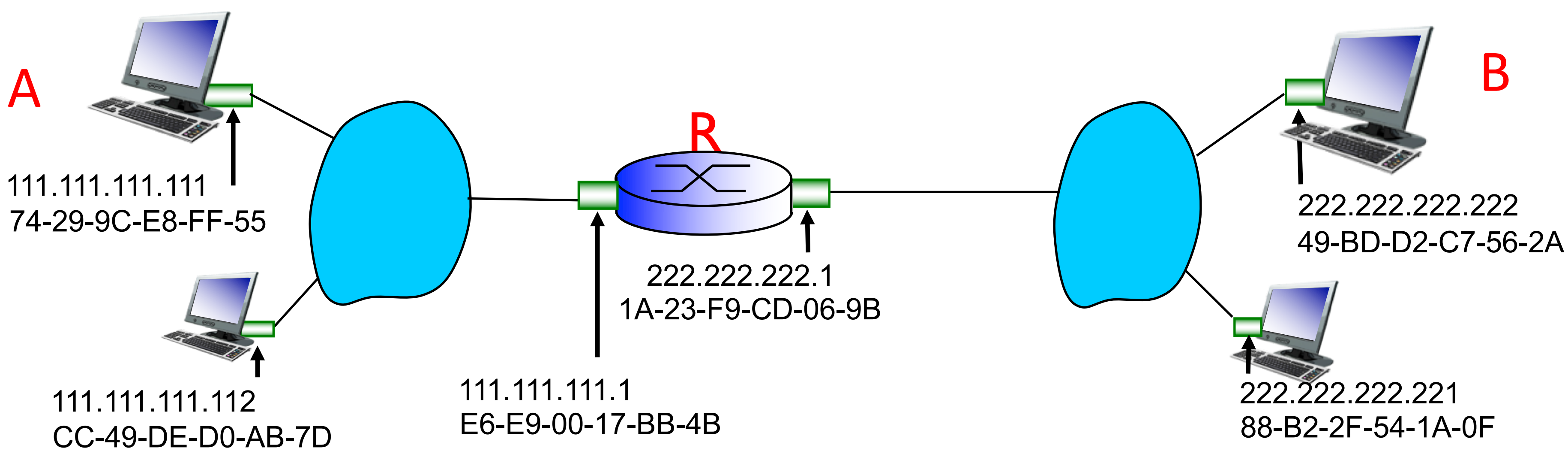
Walkthrough: send datagram from A to B via R

1. Who do we address as the IP packet destination ?
2. Who do we forward it to on the first hop?



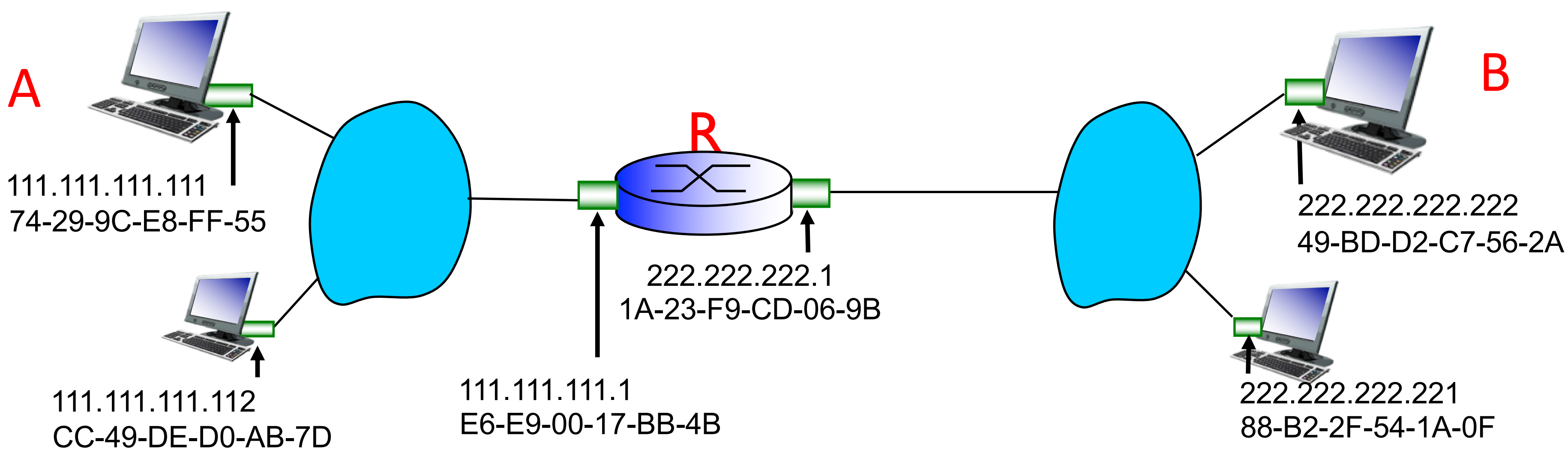
Walkthrough: send datagram from A to B via R

1. Who do we address as the IP packet destination ?
   - IP Address to B (End-to-end address to express where we want to get to)



Walkthrough: send datagram from A to B via R

2. Who do we forward it to on the first hop?
   - MAC Address to R (Intermediate address to send to router)



How does A learn the IP address of R?

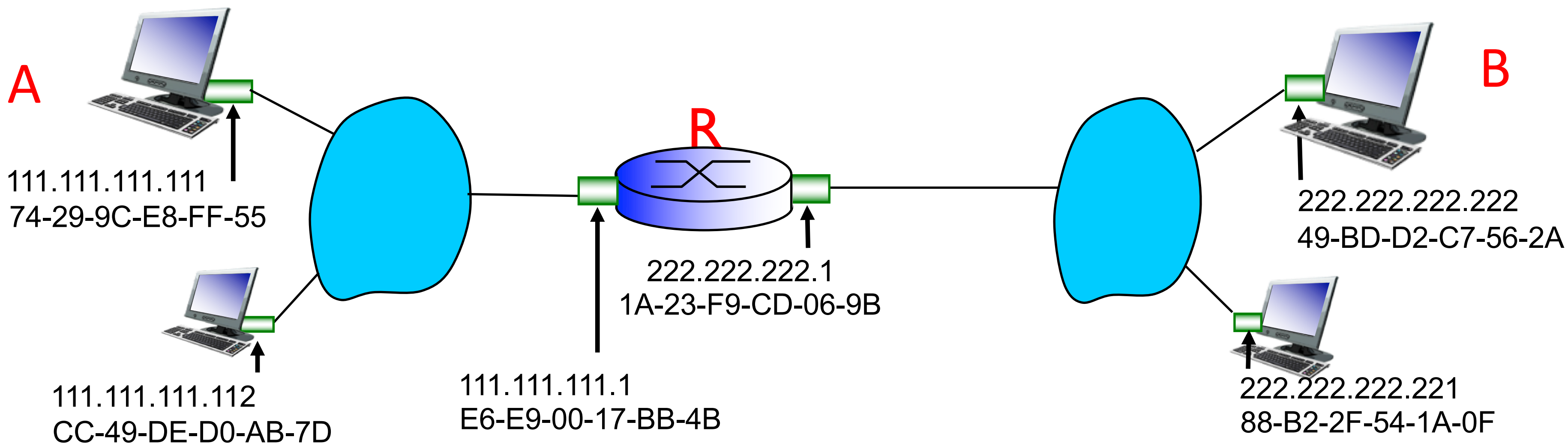
A. ARP: Address Resolution Protocol

B. DHCP: Dynamic Host Configuration Protocol

C. IP: Internet Protocol

D. Routing Protocol

why do we even need the IP address of Router ?



How does A learn the MAC address of R?

A. ARP: Address Resolution Protocol

B. DHCP: Dynamic Host Configuration Protocol

C. IP: Internet Protocol

D. Routing Protocol