Networks

Link Layer - Introduction

Adopted from material in “Computer Networking: A Top Down Approach” by Kurose and Ross and slides developed by William Conner

Link Layer - Introduction

Section 6.1

Terminology Review

- Node: host or router
- Link: communication channel that connects two adjacent nodes
  - Wired (e.g., Ethernet 802.3)
  - Wireless (e.g., Wi-Fi 802.11)
- Frame: layer-2 PDU

Link Layer

- Sits between network and physical layers
- Encapsulates IP datagrams into frames
- Single-hop data transfer over link between two adjacent nodes

Link Layer

Different link layer technologies with different services (e.g., Wi-Fi has reliable data transfer while Ethernet does not) for the same end-to-end path

Transportation Analogy

- Trip from Manhattan to Kyoto
  - Limo: Manhattan to JFK
  - Plane: JFK to Osaka
  - Train: Osaka to Kyoto

- Tourist = datagram
- Transportation segment = communication link
- Transportation mode = link layer protocol
- Travel agent = routing algorithm

Link Layer Services

- Framing
  - Encapsulate IP datagram into link layer frame
  - Frame headers contain *MAC addresses* to identify source and destination
  - MAC address ≠ IP address (more later)

- Provide medium access control (MAC) if shared medium

Link Layer Services

- Reliable delivery between adjacent nodes
  - Seldom used on low bit-error links (e.g., fiber)
  - Can be used on high bit-error links (e.g., wireless links)

Why do both link-level and end-to-end reliability make sense for wireless?

End-to-end argument violation?

Link Layer Services

- Flow control
  - Between adjacent sender and receiver
  - Sound familiar?

- Half-duplex and full duplex
  - Only one end can transmit at a time (half)
  - Both ends can transmit simultaneously (full)

Link Layer Services

- Error detection
  - Errors caused by signal attenuation and noise
  - Receiver detects presence of error
    - Signal sender for retransmission
    - Drop frame

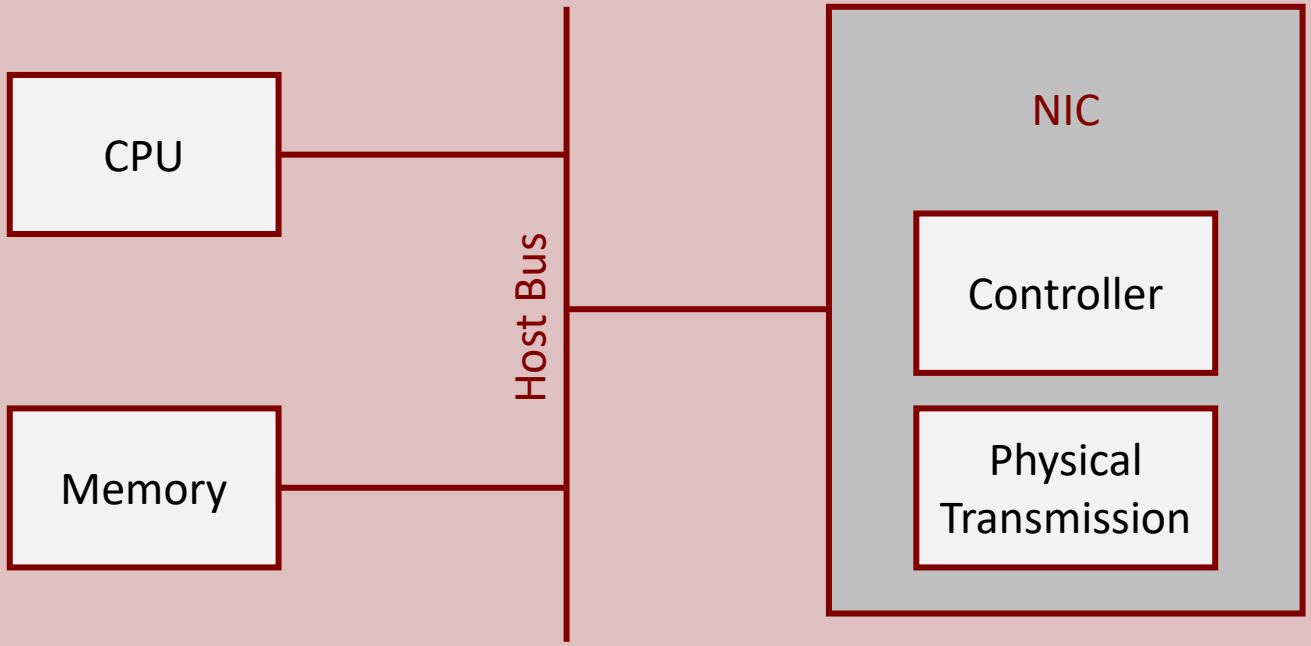
- Error correction
  - Receiver identifies *and corrects* bit errors
  - No retransmission

Adapters (NICs)

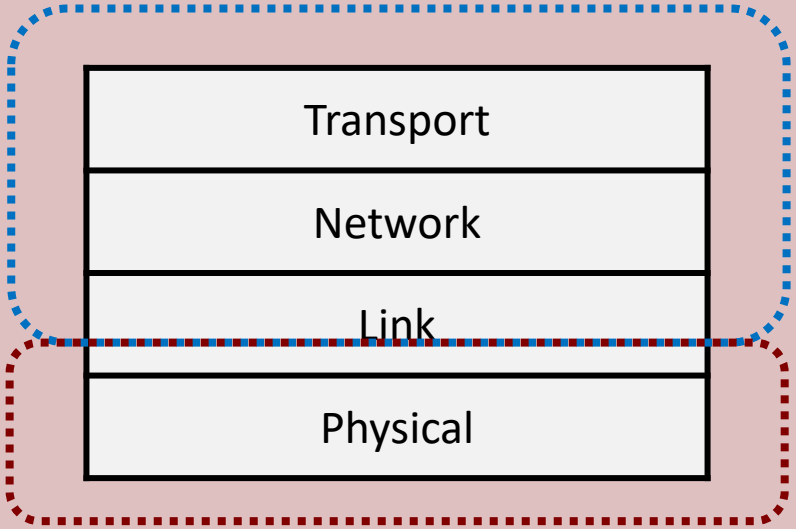
- Network interface card (NIC)
  - Ethernet card
  - 802.11 card
  - Ethernet chipset (on motherboard)

- Implements link layer and physical layer
- Component of each and every host
- Attaches into host's system bus
- Combination of hardware, software, and firmware (ROM)

Adapters (NICs)



Adapters (NICs)



Operating System

Network Adapter

Adapter (NIC) Functionality

- Error Checking
- Reliable data transmission
- Flow control
- Encapsulation and decapsulation between datagrams and frames

Thank You!

Networks

Error Detection and Correction

Adopted from material in “Computer Networking: A Top Down Approach” by Kurose and Ross and slides developed by William Conner

Error Detection and Correction

Section 6.2

Review: UDP Checksum

Goal: detect “errors” (e.g., flipped bits) in a transmitted packet

Sender

- Treat segment content (plus pseudo-header as sequence of 16-bit integers
- Checksum: addition (1’s complement of 1’s complement sum) of segment contents
- Sender puts checksum value into UDP checksum field

Receiver

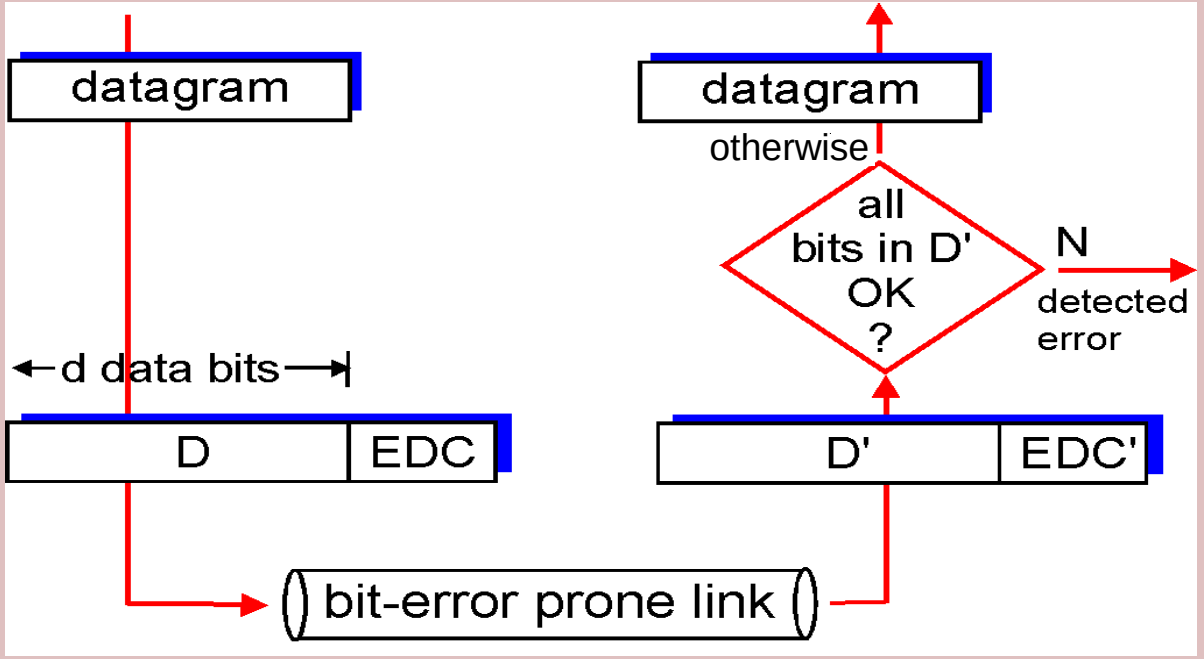
- Compute checksum of received segment
- Verify checksum to detect bit errors

Error Detection

EDC = Error Detection and Correction bits (redundancy)

D = Data protected by error checking, may include header fields

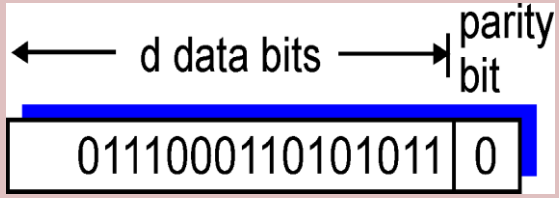
- Error detection not 100% reliable!
  - Protocol may miss some errors, but rarely
  - Larger EDC field yields better detection and correction



Parity Checking

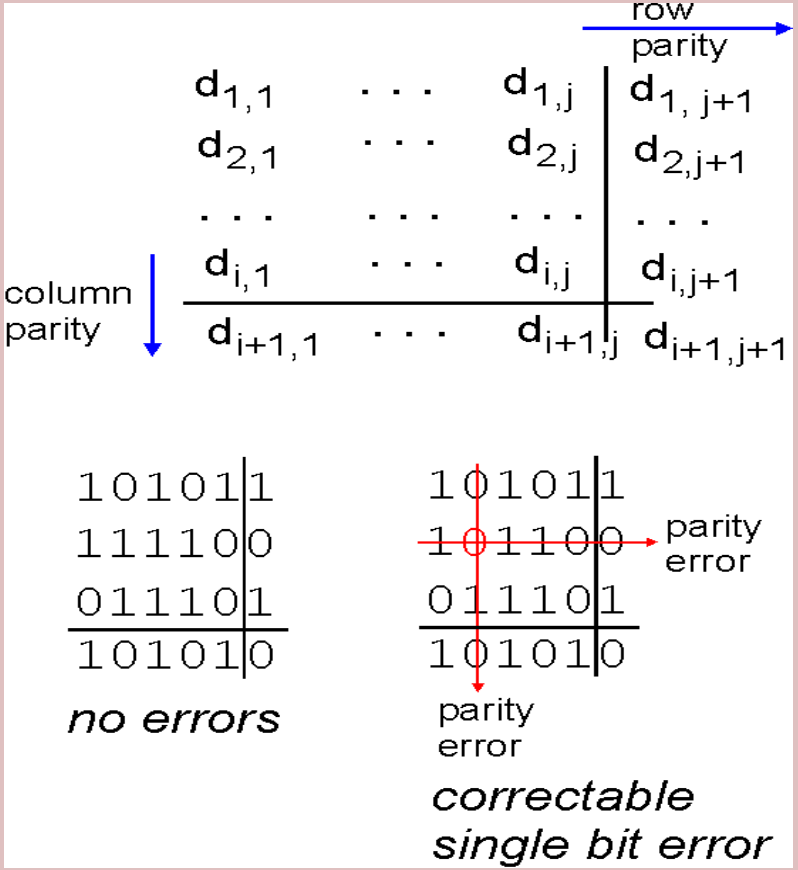
Single bit parity:

- Detect single bit errors



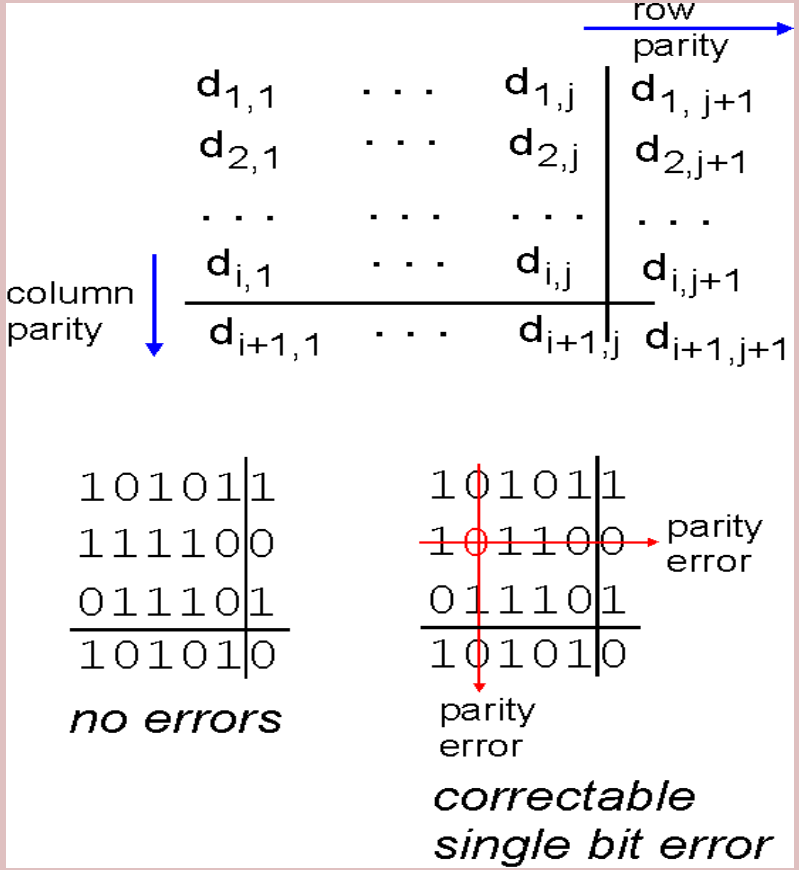
Two-dimensional bit parity:

- Detect and correct single bit errors



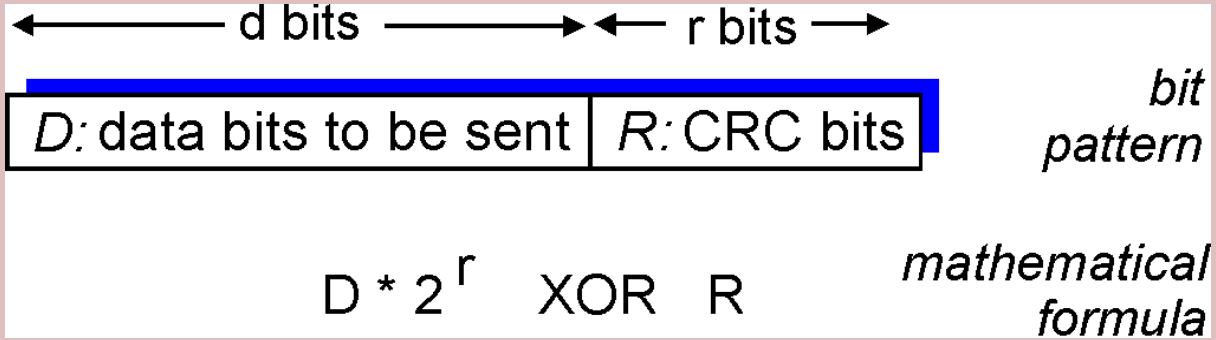
Parity Checking

Can errors go undetected with 2-d parity?



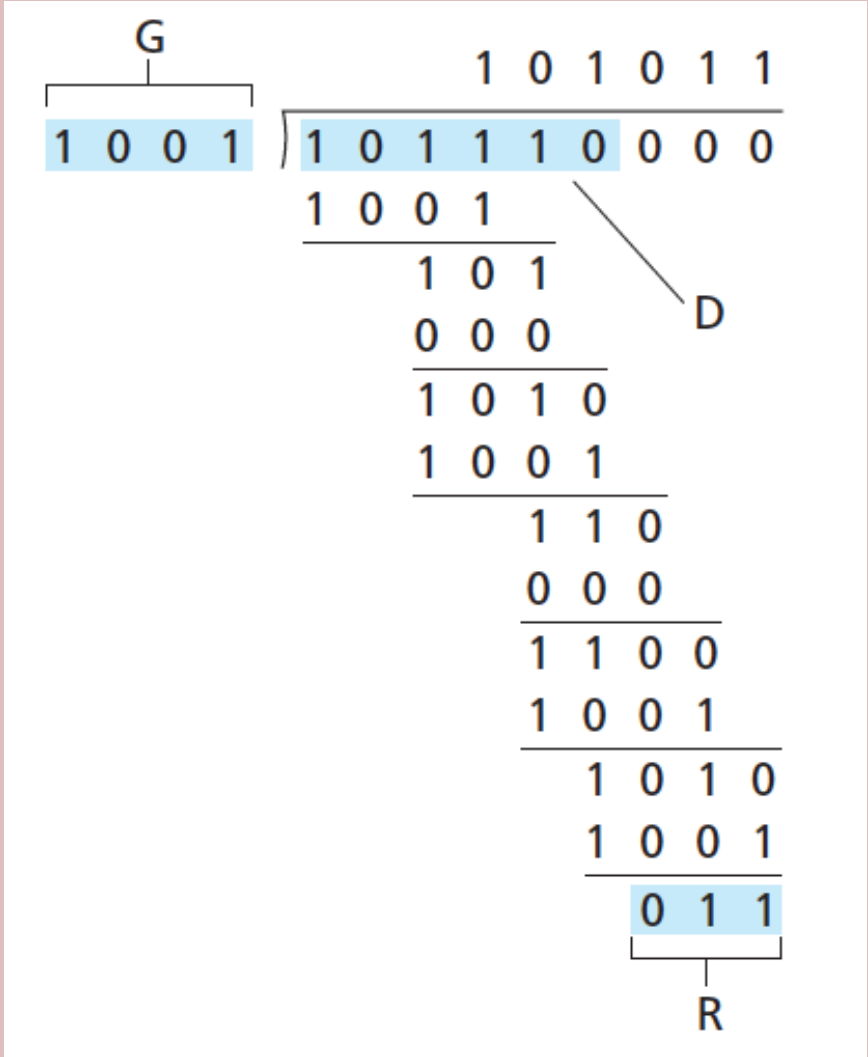
Cyclic Redundancy Check

- More powerful error-detection coding
- View data bits, D, as a binary number
- Choose r+1 bit pattern (generator), G (starts with 1)
- Goal, choose r CRC bits, R, such that
  - <D,R> exactly divisible by G (mod 2)
  - Receiver knows G, divides <D,R> by G. If non-zero remainder, error detected!
  - Can detect all burst errors less than r+1 bits
  - Can detect any *odd* number of isolated errors
- Widely used in practice (Ethernet, 802.11 WiFi, ATM)



CRC Example

$R = \text{remainder}[D \cdot 2^r \div G]$



Thank You!

Networks

Multiple Access Links and Protocols

Adopted from material in “Computer Networking: A Top Down Approach” by Kurose and Ross and slides developed by William Conner

Multiple Access Links and Protocols

Section 6.3

Link Layer Protocols

- Point-to-point (dedicated) links
  - PPP for dial-up access
  - Point-to-point link between Ethernet switch and host
- Broadcast (shared) link
  - Wired Ethernet (traditional bus topology)
  - 802.11 Wireless LAN

Broadcast Link

- Single shared channel
- Interference caused by two or more simultaneous transmissions
- Collision occurs if a node receives two or more signals at the same time

Multiple Access Protocols

- Distributed algorithm that determines how nodes share a channel (i.e., determine when a node can transmit)

- Communication about channel sharing must also use the shared channel (i.e., no out-of-band communication)!

Multiple Access Protocols

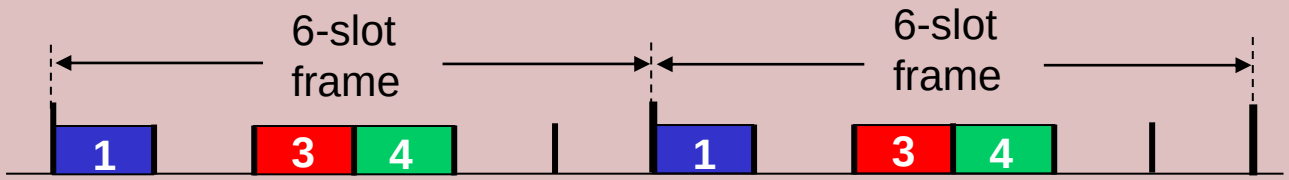
- Assume broadcast channel with rate *R* bps
- Desired properties of an ideal protocol
  - When one node wants to transmit, it can transmit at rate *R*
  - When *M* nodes want to transmit, each can send at an average rate of *R/M*
  - Fully decentralized
    - No special node to coordinate transmissions
    - No synchronization of clocks, slots
  - Simple

Medium Access Control (MAC) Protocols

- Channel partitioning
  - Divide channel (time slots, frequency)
  - Allocate a piece to each node for exclusive use
- Random access
  - Channel not divided (collisions allowed)
  - *Recover* from collisions
- Taking turns
  - Nodes take turns
  - Nodes with more to send can take longer turns

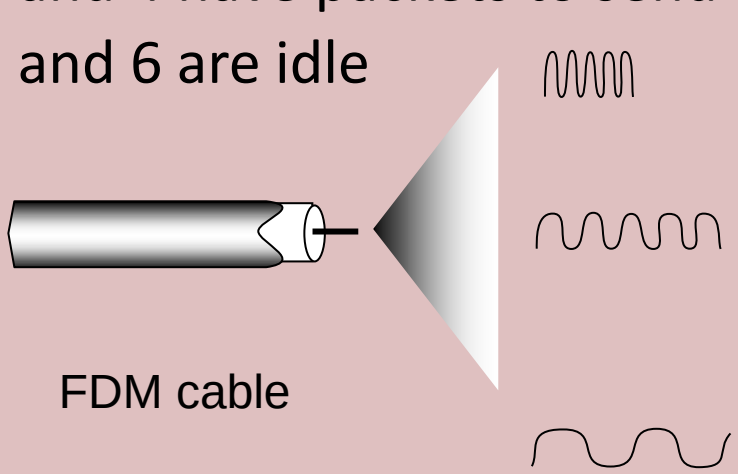
Time Division Multiple Access (TDMA)

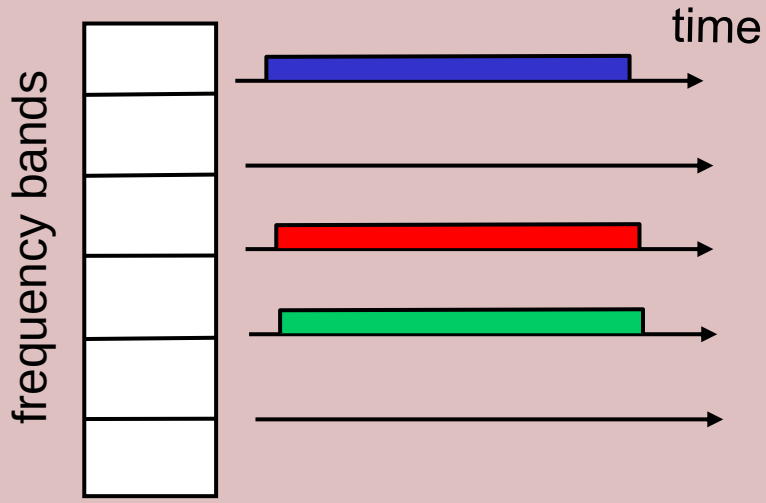
- Access channel in *rounds*
- Each station gets a fixed length slot in each round
- Unused slots go idle (inefficient)
- Example
  - 6-node LAN
  - 1, 3, and 4 have packets to send
  - 2, 5, and 6 are idle



Frequency Division Multiple Access (FDMA)

- Channel spectrum divided into frequency bands
- Each station assigned fixed frequency band
- Unused transmission time in frequency bands goes idle
- Example
  - 6-node LAN
  - 1, 3, and 4 have packets to send
  - 2, 5, and 6 are idle





Random Access Protocols

- When a node has a packet to send
  - Transmit at full channel data rate *R*
  - No *a priori* coordination among nodes

- 2+ transmitting nodes ⇒ collision

- Random access protocol specifies:
  - How to detect collisions
  - How to recover from collisions (e.g., via delayed retransmissions)

Random Access Protocols

- Slotted ALOHA
- Pure ALOHA
- Carrier Sensing Multiple Access (CSMA)
  - CSMA
  - CSMA/CD
  - CSMA/CA (next week)

Slotted ALOHA

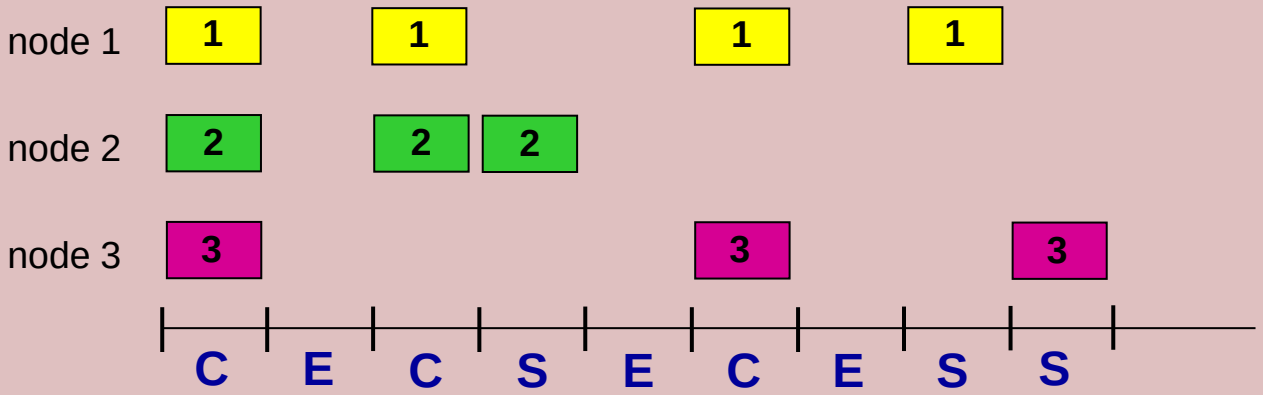
Assumptions:

- All frames same size
- Time divided into equal size slots (time to transmit 1 frame)
- Nodes start to transmit only at slot beginning
- Nodes are synchronized
- If 2 or more nodes transmit in a slot, all nodes detect collision

Operation:

- When a node obtains a fresh frame, transmits in the next slot
  - *If no collision:* node can send a new frame in the next slot
  - *If collision:* node retransmits frame in each subsequent slot with probability *p* until success

Slotted ALOHA



Pros:

- Single active node can continuously transmit at full rate of channel
- Highly decentralized: only slots in nodes need to be in sync
- Simple

Cons:

- Collisions, wasting slots
- Idle slots
- Nodes may be able to detect collision in less than the time to transmit a packet, wasting the remaining slot
- Clock synchronization

Slotted ALOHA Efficiency

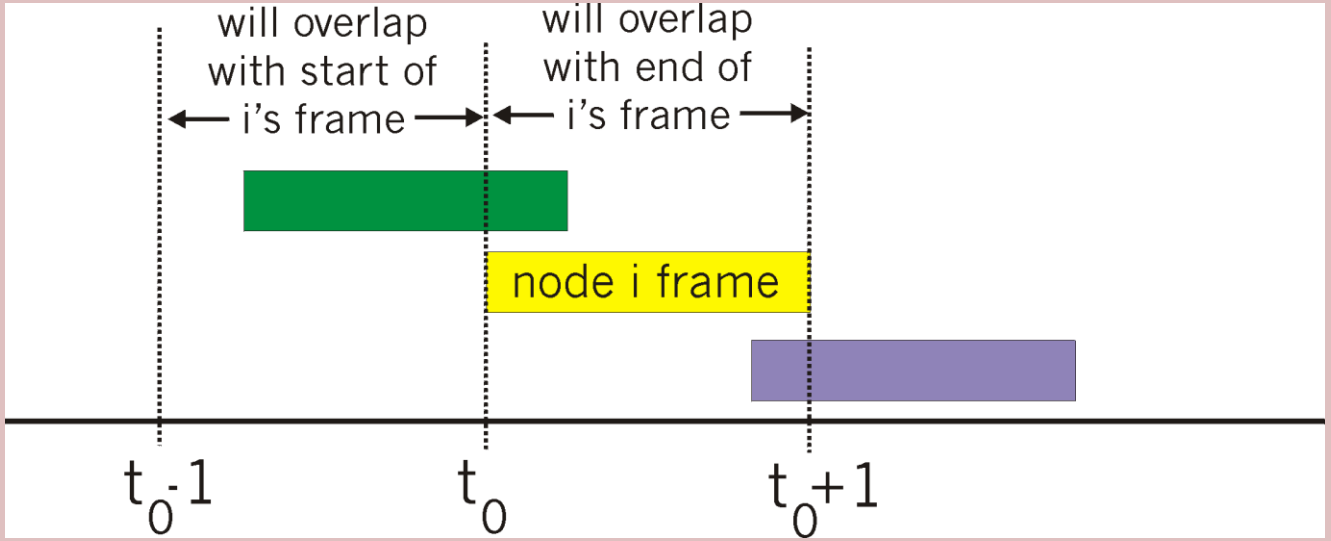
- *Efficiency:* long-run fraction of successful slots (many nodes, all with many frames to send)
- *Suppose: N* nodes with many frames to send, each transmits in slot with probability *p*
- Probability that a given node has success in a slot = $p(1-p)^{N-1}$
- Probability that *any* node has a success = $Np(1-p)^{N-1}$
- Maximum efficiency: find p^* that maximizes $Np^*(1-p^*)^{N-1}$
- For many nodes, take the limit as *N* goes to infinity, gives:

  Max efficiency = 1/e ≈ 0.37

- *At best:* Channel used for useful transmissions about 37% of the time!

Pure (Unslotted) ALOHA

- Simpler, no synchronization
- When frame first arrives
  - Transmit immediately
- Collision probability increases:
  - Frame sent at t_0 collides with other frames sent in $(t_0-1, t_0+1]$



Pure ALOHA Efficiency

P(success by given node) = P(node transmits)·
P(no other node starts to transmit in $(t_0-1, t_0]$)·
P(no other node starts to transmit in $[t_0, t_0+1)$)

$= p\cdot(1-p)^{N-1}\cdot(1-p)^{N-1}$

$= p\cdot(1-p)^{2(N-1)}$

Choosing optimum p and then letting n→∞

$= 1/(2e) \approx 0.18$

Even *worse* than slotted ALOHA!

Carrier Sense Multiple Access (CSMA)

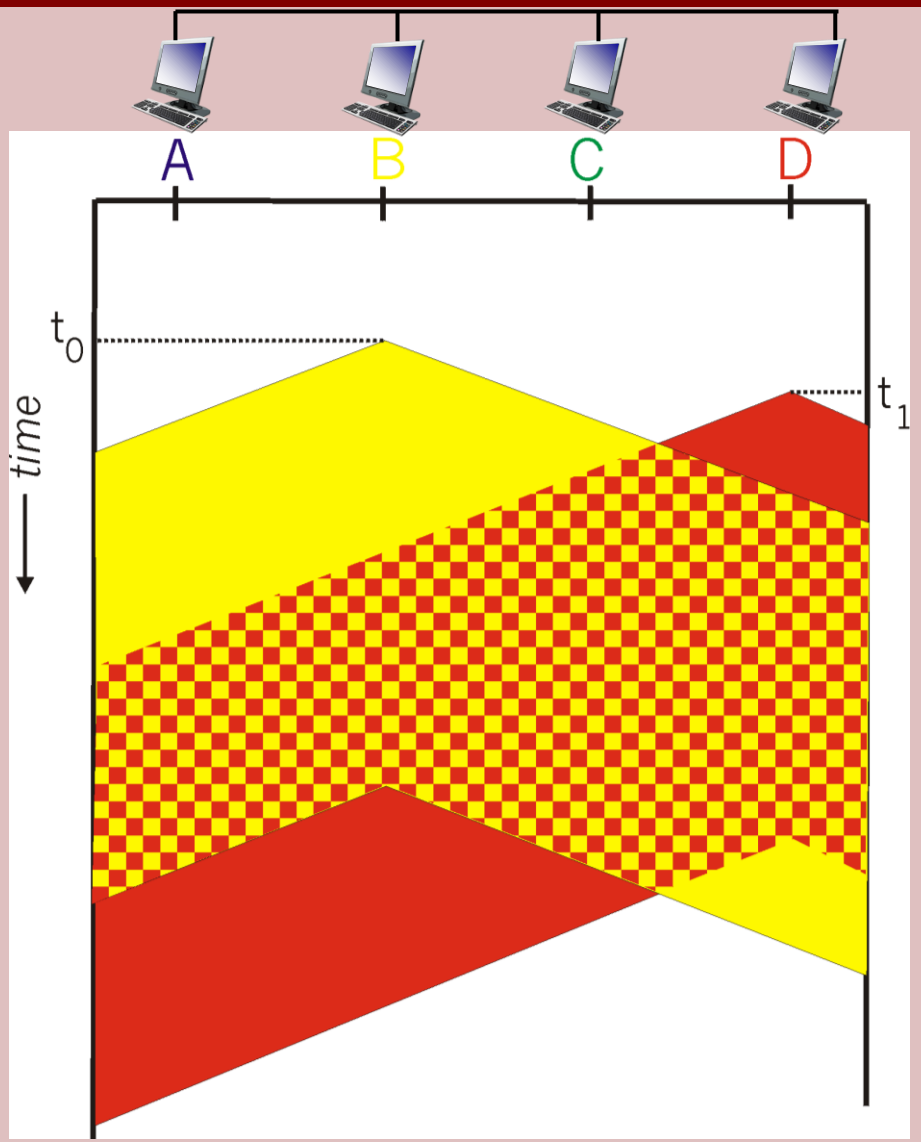
- Listen before transmit
- If channel sensed idle, transmit entire frame
- If channel sensed busy, defer transmission
- Human analogy: don’t interrupt others

Is that sufficient?

CSMA: Collisions

- Collisions *can* still occur: propagation delay means that it's possible that two nodes might not hear each other's transmission

- Collision: entire packet transmission time wasted
  - Distance and propagation delay play a role in determining collision probability



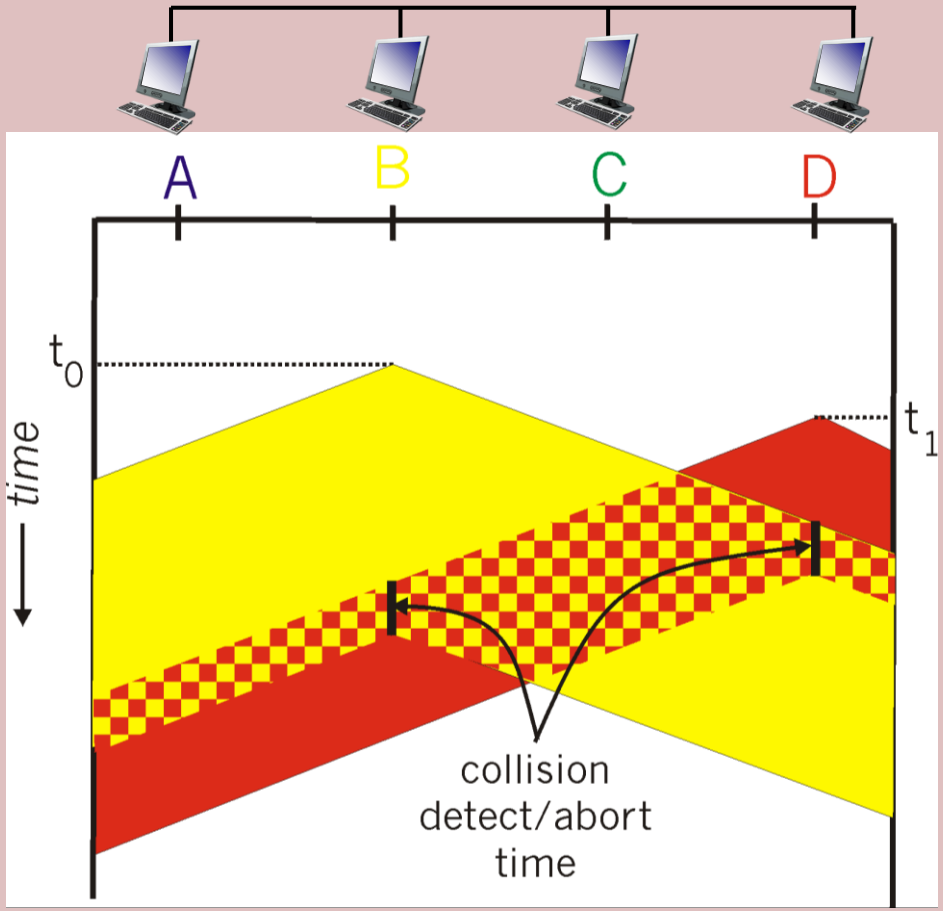
CSMA/CD: Collision Detection

- Carrier sensing, deferral as in CSMA
  - Collisions *detected* within short time
  - Colliding transmissions aborted, reducing channel wastage

- Collision detection:
  - Easy in wired LANs: measure signal strengths, compare transmitted and received signals
  - Difficult in wireless LANs: received signal strength is overwhelmed by local transmission strength

- Human analogy: the polite conversationalist

CSMA/CD: Collision Detection



CSMA/CD Algorithm

1. NIC receives datagram from network layer, creates frame.
2. If NIC senses channel is idle, starts frame transmission. If NIC senses channel is busy, waits until channel is idle, then transmits.
3. If NIC transmits entire frame without detecting another transmission, NIC is done with the frame!
4. If NIC detects another transmission while transmitting, aborts and sends jam signal
5. After aborting, NIC enters *binary (exponential) backoff*:
   - After *m*th collision, NIC chooses *K* at random from $\{0, 1, 2, \ldots, 2^m-1\}$
     - NIC waits $K \cdot 512$ bit times, returns to Step 2
   - *Longer backoff interval with more collisions*

CSMA/CD Efficiency

- t_{prop} = max prop delay between 2 nodes in LAN
- t_{trans} = time to transmit max-size frame

$$\text{efficiency} = \frac{1}{1 + 5\frac{t_{prop}}{t_{trans}}}$$

- Efficiency goes to 1:
  - As t_{prop} goes to 0
  - As t_{trans} goes to infinity
- Better performance than ALOHA: and simple, cheap, decentralized!

Review: MAC Protocols

Channel partitioning MAC protocols:

- Share channel efficiently and fairly at high load
- Inefficient at low load: delay in channel access, *1/N* bandwidth allocated even if only 1 active node!

Random access MAC protocols

- Efficient at low load: single node can fully utilize channel
- High load: collision overhead

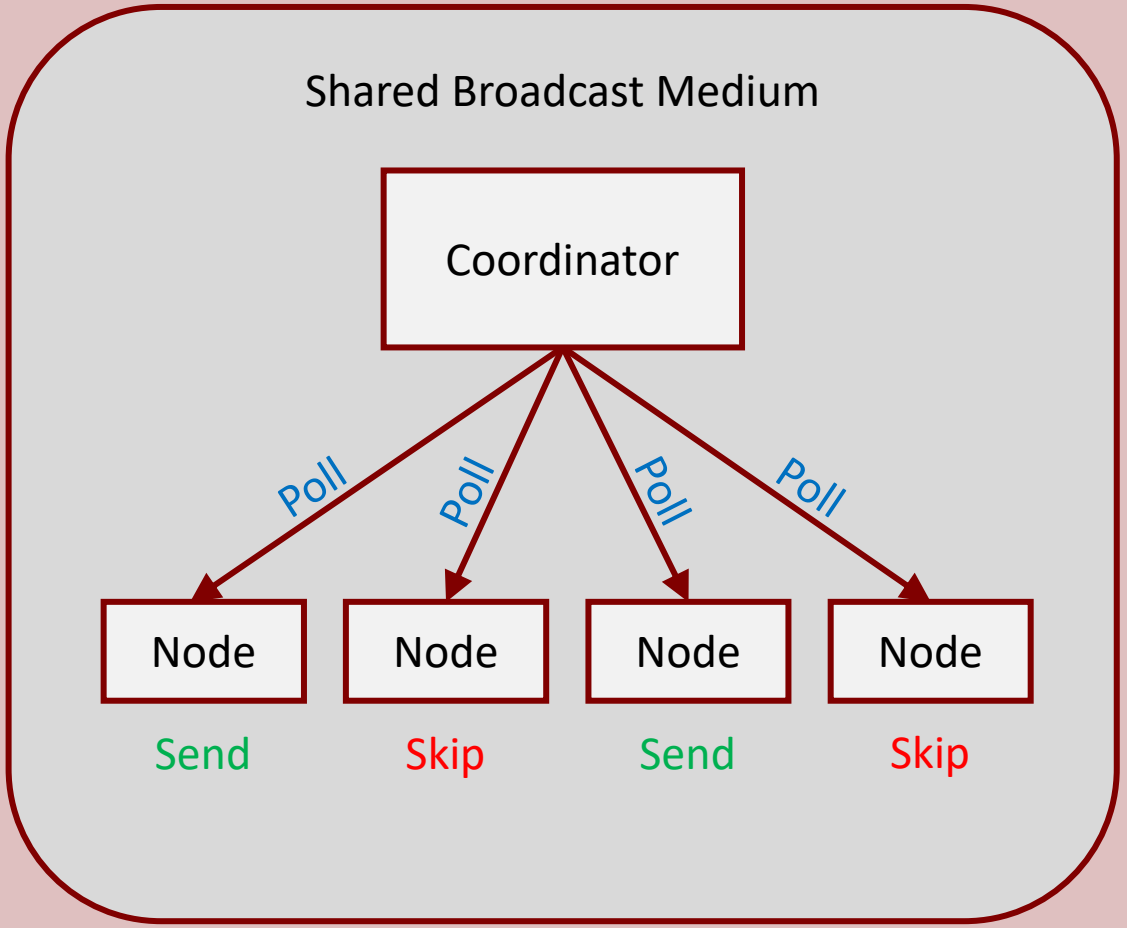
"Taking turns" protocols

- Look for the best of both worlds!

Taking Turns Protocols

Polling:

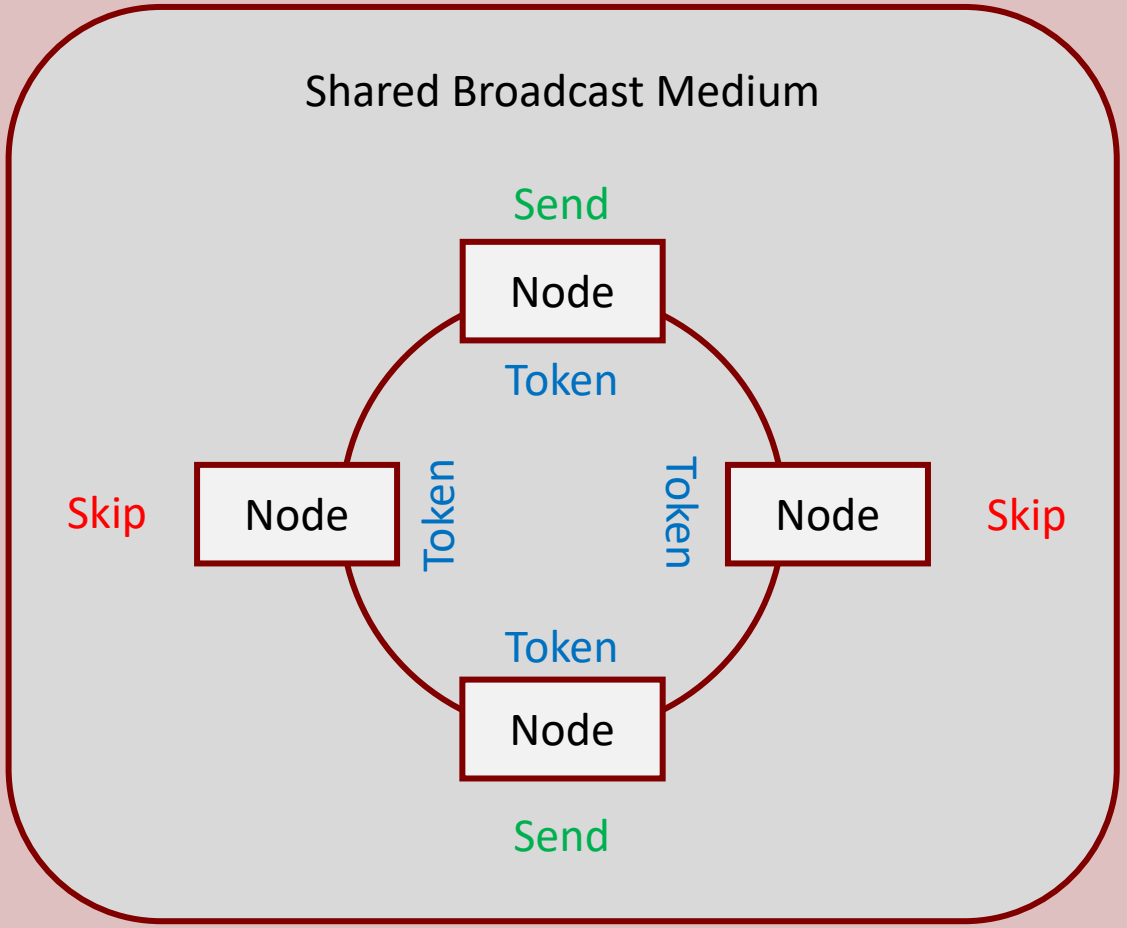
- Coordinator “invites” nodes to transmit in turn
- Concerns:
  - Polling overhead
  - Latency
  - Single point of failure (coordinator)



Taking Turns Protocols

Token passing:

- Control *token* passed from one node to the next sequentially.
- Token message
- Concerns:
  - Token overhead
  - Latency
  - Single point of failure (token)



Summary of MAC protocols

- *Channel partitioning,* by time or frequency
  - Time Division (TDMA)
  - Frequency Division (FDMA)

- *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

- *Taking turns*
  - Polling from central site, token passing
  - Token ring

Thank You!

Networks

Local Area Networks

Adopted from material in “Computer Networking: A Top Down Approach” by Kurose and Ross and slides developed by William Conner

Local Area Networks

Section 6.4

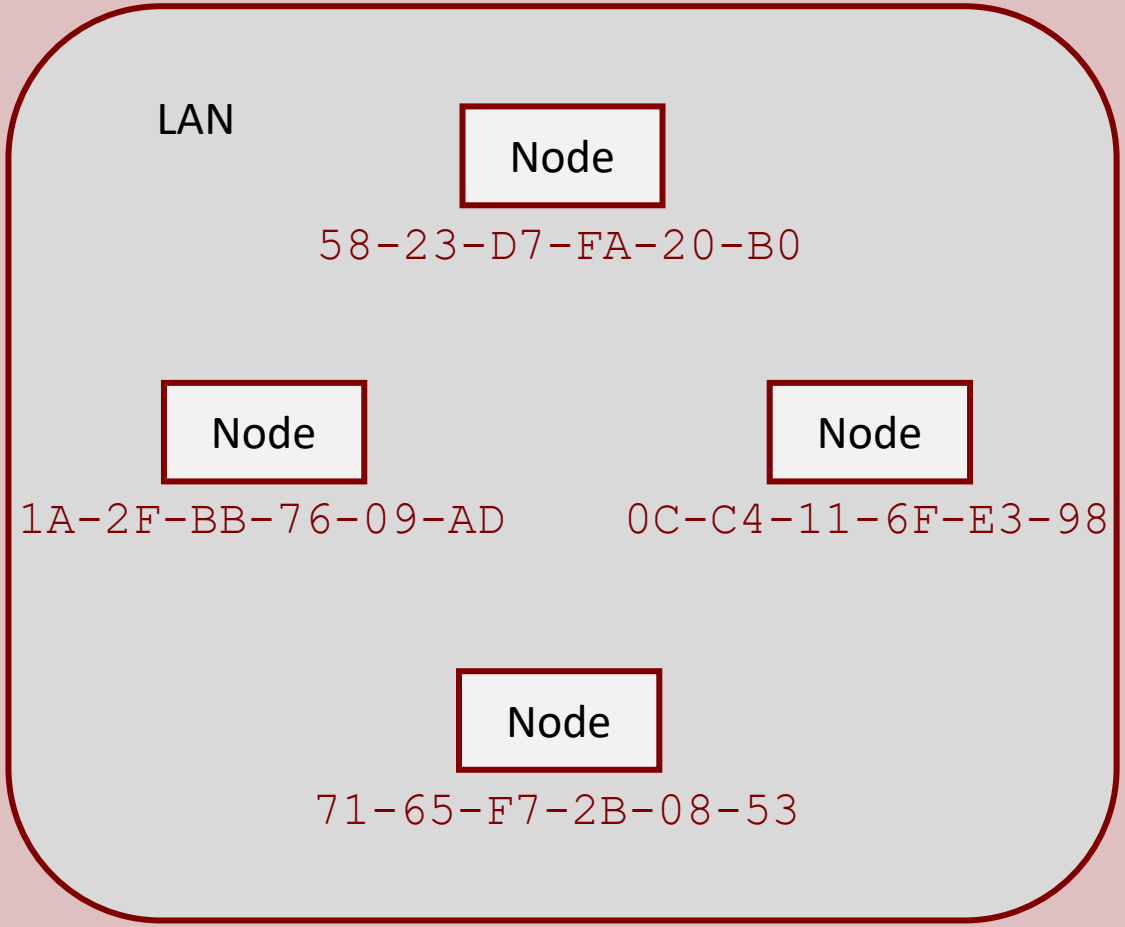
MAC Addresses

- 32-bit IP address:
  - *Network-layer* address for interface
  - Used for layer 3 (network layer) forwarding

- MAC (or LAN or physical or Ethernet) address:
  - *Used "locally" to get frame from one interface to another physically-connected interface (same network, in an IP-addressing sense)*
  - 48-bit MAC address (for most LANs) burned in NIC ROM, also sometimes software settable
  - E.g.: 1A-2F-BB-76-09-AD
  - Hexadecimal (base 16) notation (each "numeral" represents 4 bits)

MAC Addresses

Each adapter on LAN has unique *MAC* address

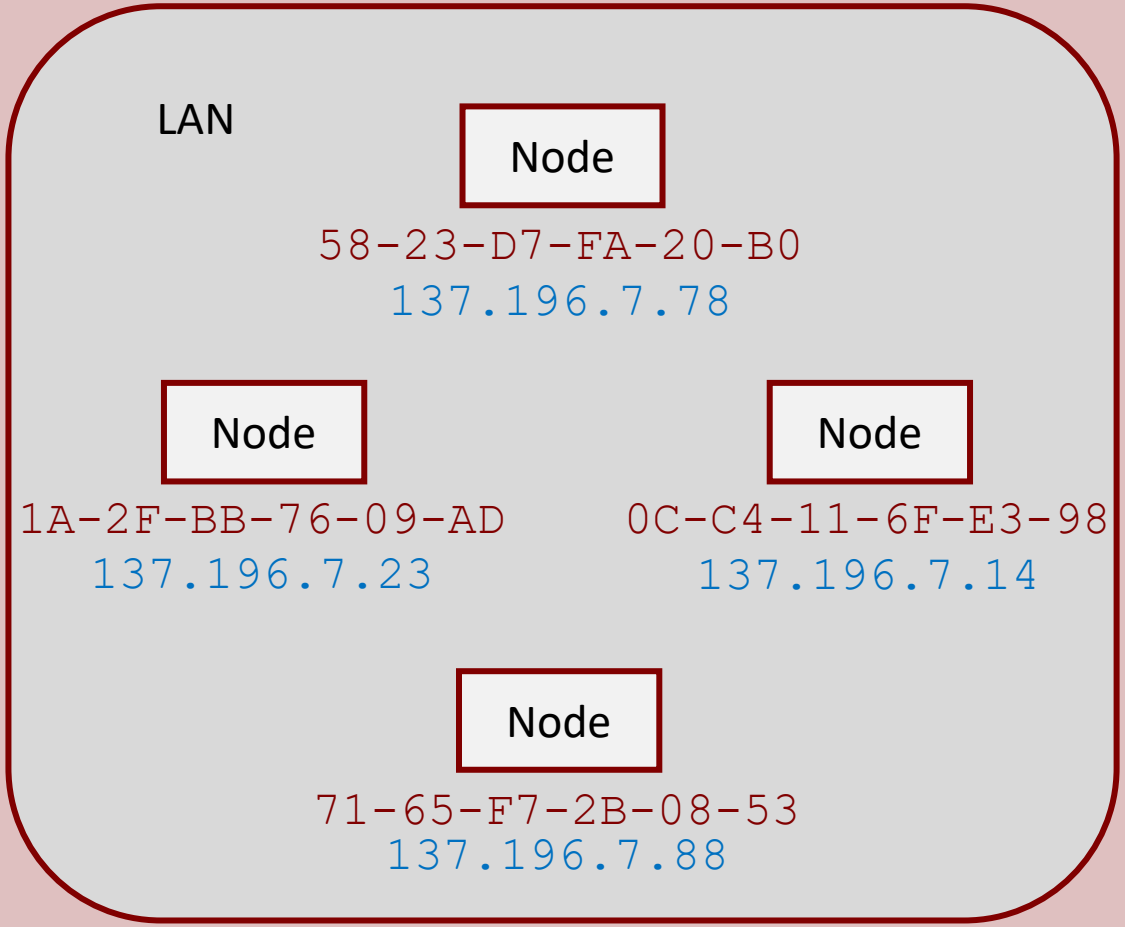


MAC Addresses

- MAC address allocation administered by IEEE
- Manufacturer buys portion of MAC address space (to assure uniqueness)
- Analogy:
  - MAC address: like Social Security Number
  - IP address: like postal address
- MAC flat address → portability
  - Can move LAN card from one LAN to another
- IP hierarchical address *not* portable
  - Address depends on IP subnet to which node is attached

Address Resolution Protocol (ARP)

Given an IP address, how can we determine an interface's MAC address?



ARP

- Address Resolution Protocol (ARP) resolves 32-bit IPv4 addresses to 48-bit MAC addresses
- ARP is plug-and-play (no manual configuration needed)
- ARP messages are encapsulated in link layer frames

ARP Table

- Maps LAN nodes' IP addresses to MAC addresses

<IP address, MAC address, TTL>

- Addresses are forgotten after time-to-live (TTL), which is typically 20 minutes

Example: ARP

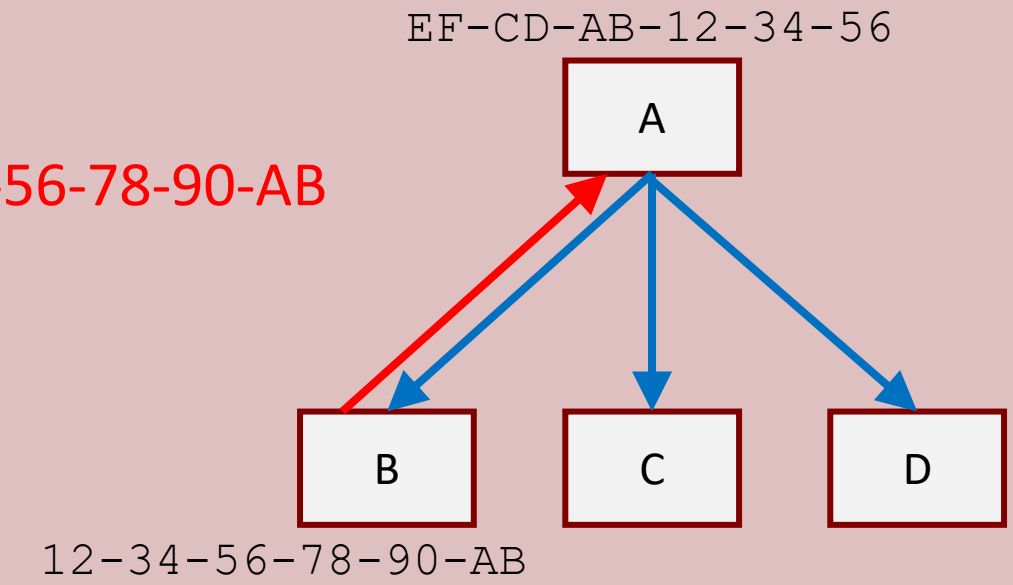
Suppose A wants to send an IP datagram to B, but does not know B's MAC address

ARP Query

- What is B's MAC address?
- Destination: FF-FF-FF-FF-FF-FF

ARP Response

- I am B and my MAC address is 12-34-56-78-90-AB
- Destination EF-CD-AB-12-34-56



ARP Table update at A

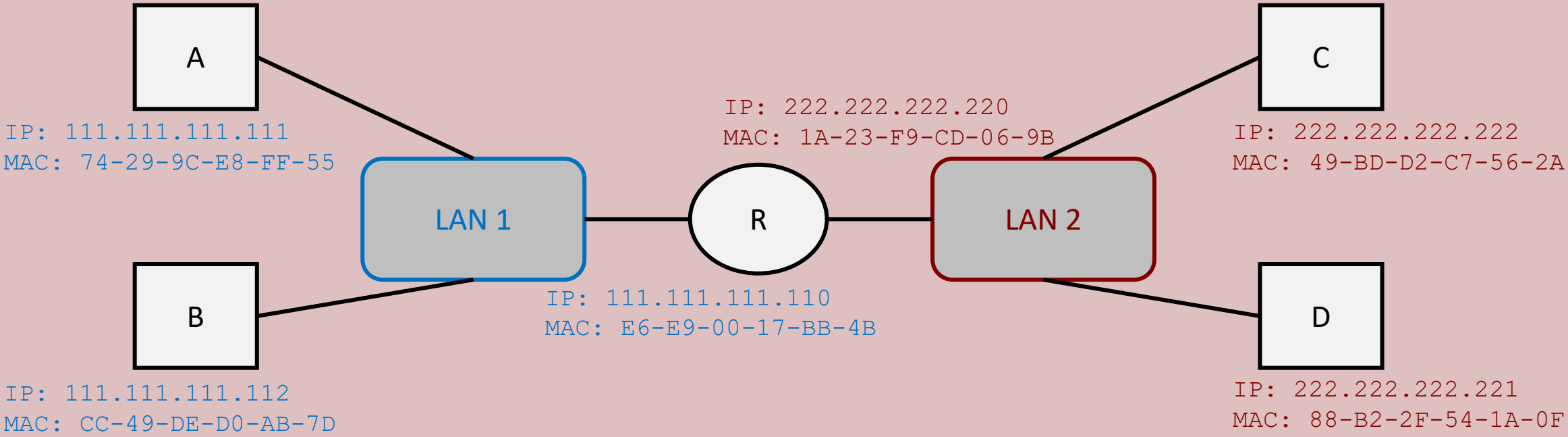
- IP: B, MAC: 12-34-56-78-90-AB

Routing Between LANs

Scenario: *A* wants to send IP datagram to *C*

A knows *C*'s IP address, but not *C*'s MAC address

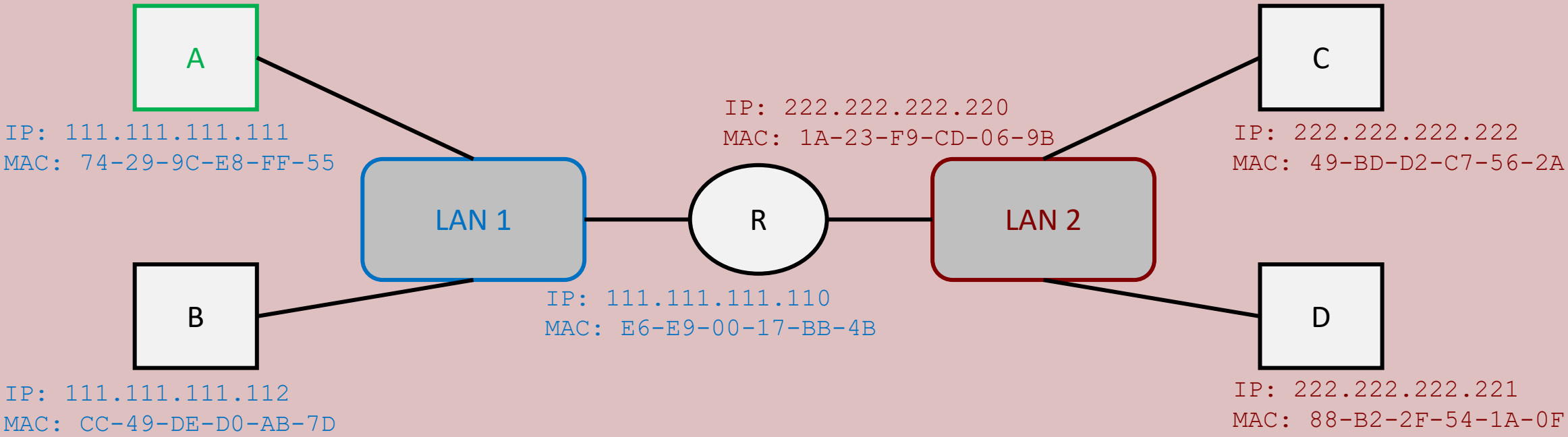
A knows router *R*'s IP and MAC addresses (how?)



Routing Between LANs

A creates IP datagram (src: *A*'s IP, dst: *C*'s IP)

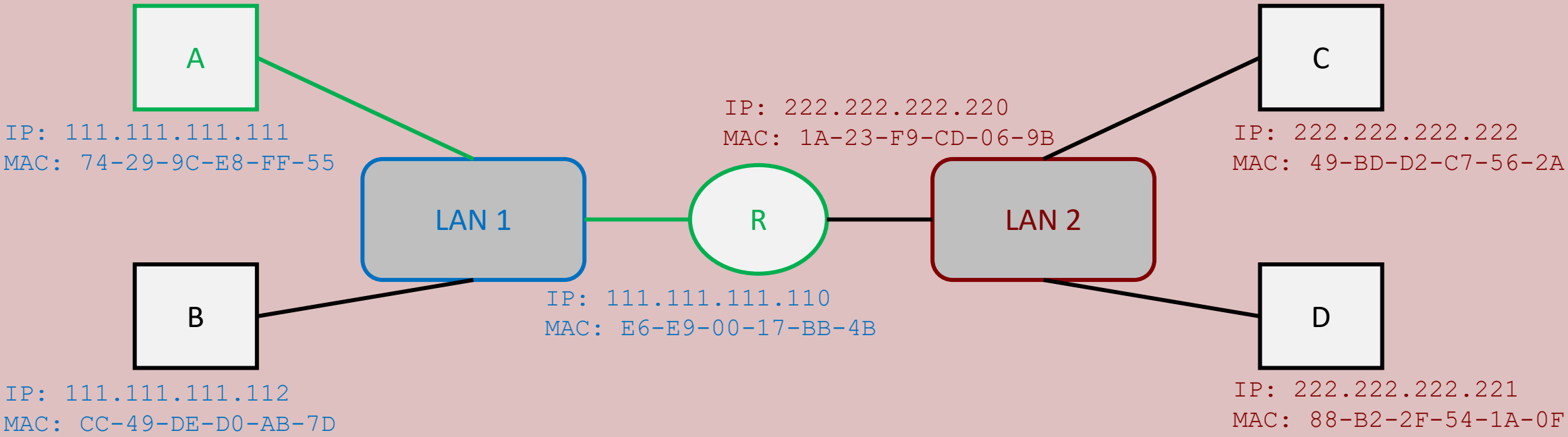
A encapsulates IP datagram into frame (src: *A*'s MAC, dst: *R*'s MAC)



Routing Between LANs

A sends frame to *R*

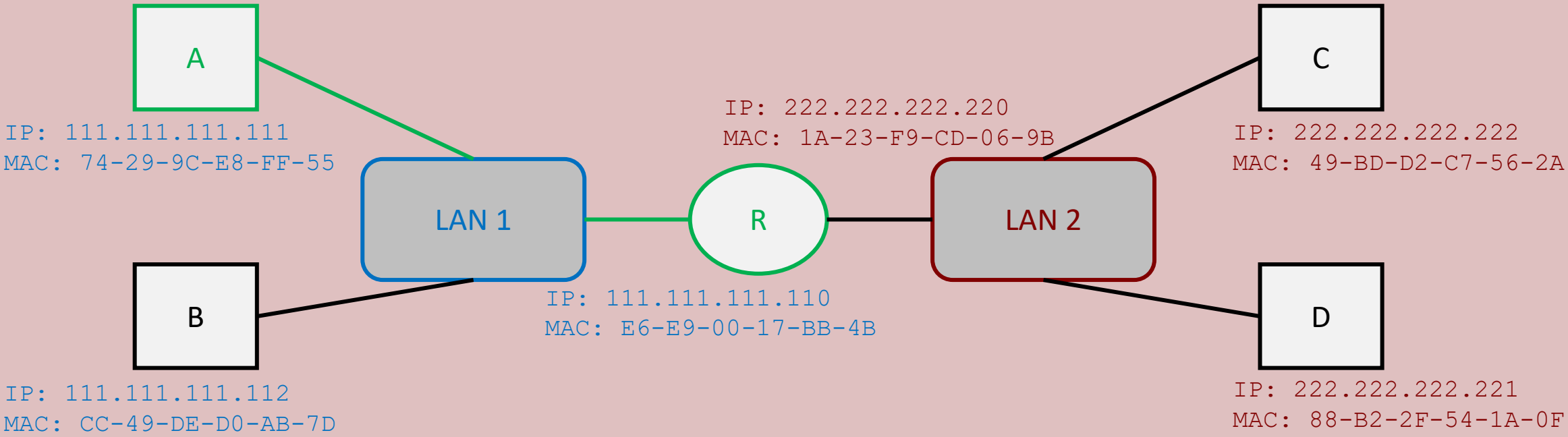
R decapsulates IP datagram from frame



Routing Between LANs

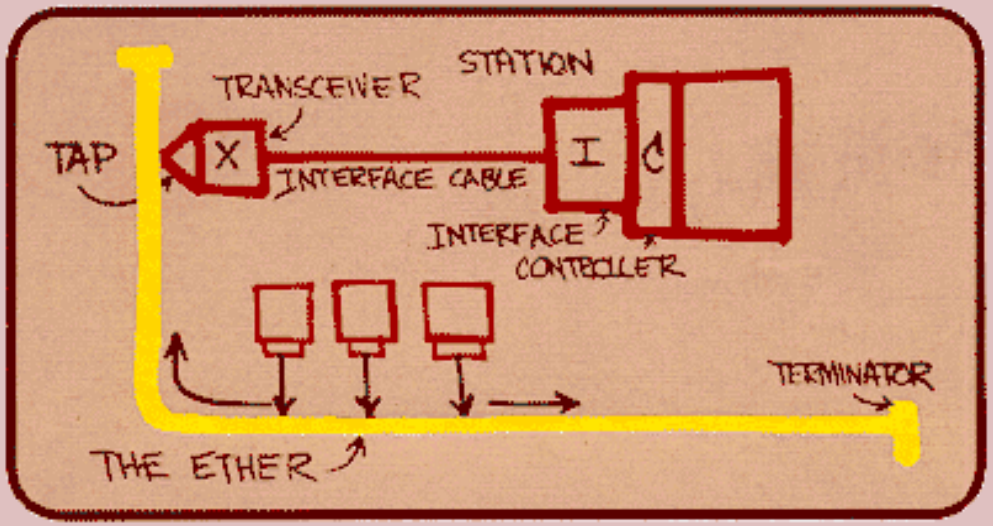
R forwards IP datagram (src: *A*'s IP, dst: *C*'s IP)

R encapsulates IP datagram into frame (src: *R*'s MAC, dst: *C*'s MAC)



Ethernet

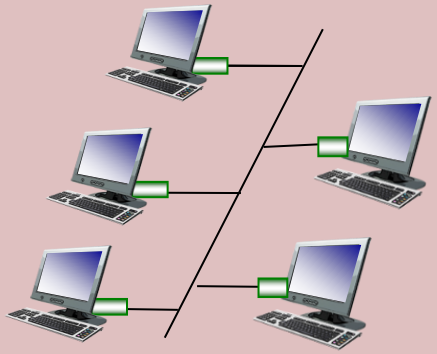
- "Dominant" wired LAN technology
  - Single chip, multiple speeds (e.g., Broadcom BCM5761)
  - First widely used LAN technology
  - Simpler, cheap
  - Kept up with speed race: 10 Mbps – 10 Gbps



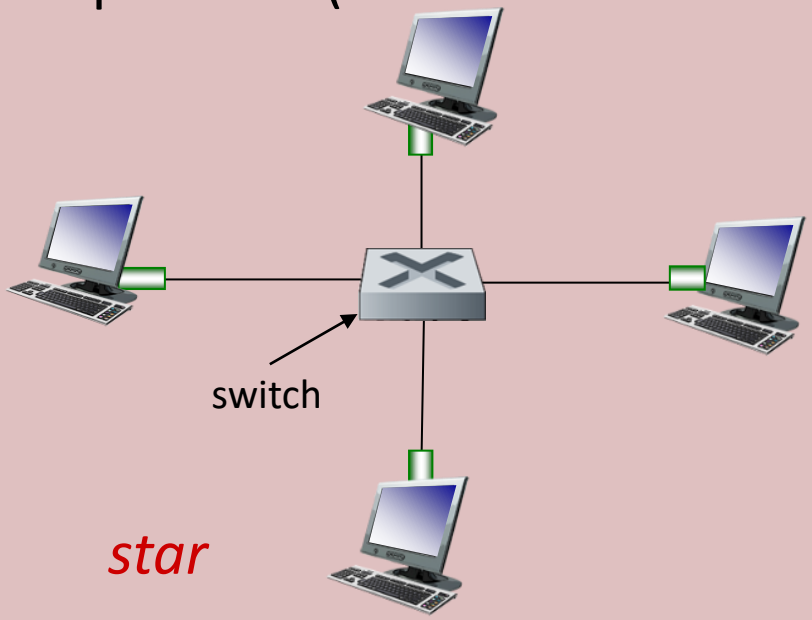
Metcalfe's Ethernet sketch

Ethernet: Physical Topology

- *Bus:* popular through mid 90s
  - All nodes in same collision domain (can collide with each other)
- *Star:* prevails today
  - Active *switch* in center
  - Each "spoke" runs a (separate) Ethernet protocol (nodes do not collide with each other)



bus: coaxial cable



star

Ethernet Frame Format

Ethernet Frame Format

- Preamble: 7 bytes of `0x10` followed by SFD byte `0x11`; used for clock recovery
- Addresses: source and destination 48-bit MAC addresses
- Type: upper layer protocol (typically IP)
- CRC: 32-bit cyclic redundancy check

Ethernet Protocol

- Connectionless: no handshaking between sending and receiving NICs
- Unreliable: receiving NIC doesn’t send ACKs or NACKs to sending NIC
- Ethernet’s MAC protocol: unslotted CSMA/CD with binary backoff

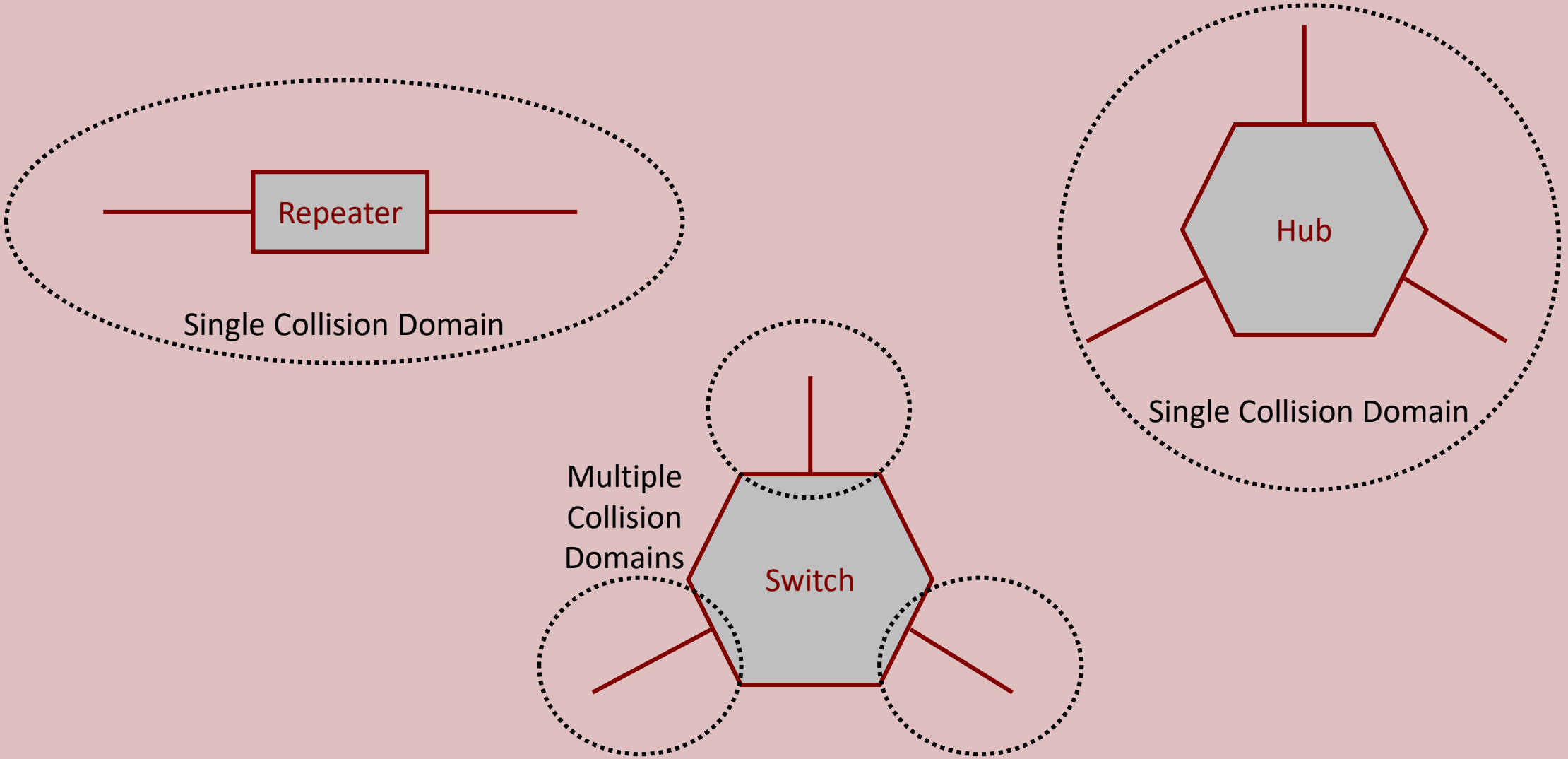
Ethernet Standards

- Many challengers have come and gone
- Ethernet has been updated many times over the years to defeat challengers
- Ethernet standard (IEEE 802.3)
  - Same MAC protocol and frame format
  - Different speeds: 2 Mbps, 10 Mbps, 100 Mbps, 1 Gbps, 10 Gbps, 40 Gbps
  - Different physical media: copper, cable, fiber

Ethernet Switch

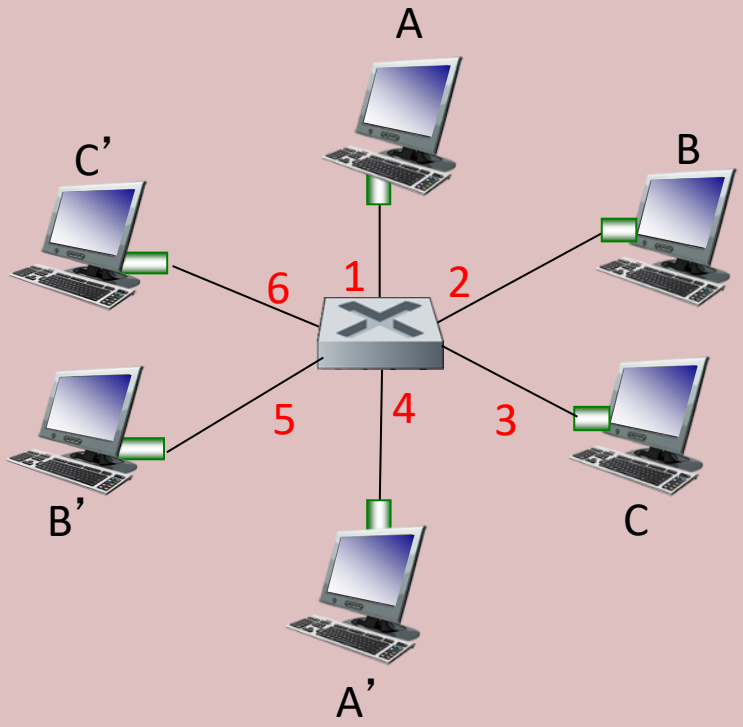
- Link layer device for Ethernet
- Stores and forwards frames based on destination MAC address (familiar?)
- Transparent to attached hosts
- Self-learning (no manual configuration)

Ethernet Devices



Switch: *Multiple* Simultaneous Transmissions

- Hosts have dedicated, direct connection to switch
- Switches buffer frames
- Ethernet protocol used on *each* incoming link, but no collisions; full duplex
- *Switching:* A-to-A' and B-to-B' can transmit simultaneously, without collisions

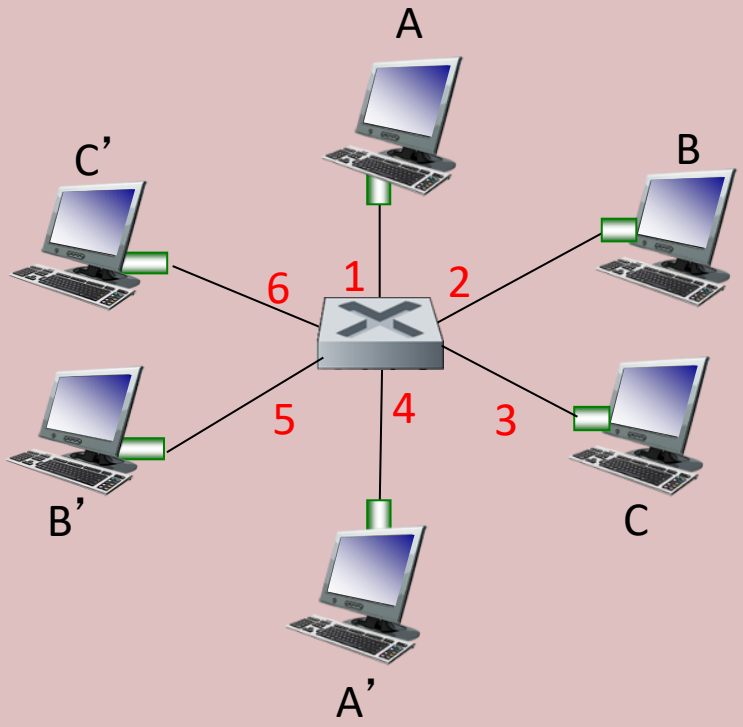


switch with six interfaces (1,2,3,4,5,6)

Switch Forwarding Table

- How does switch know A' reachable via interface 4, B' reachable via interface 5?
  - Each switch has a switch table, each entry:
    - (MAC address of host, interface to reach host, time stamp)
    - Looks like a routing table!

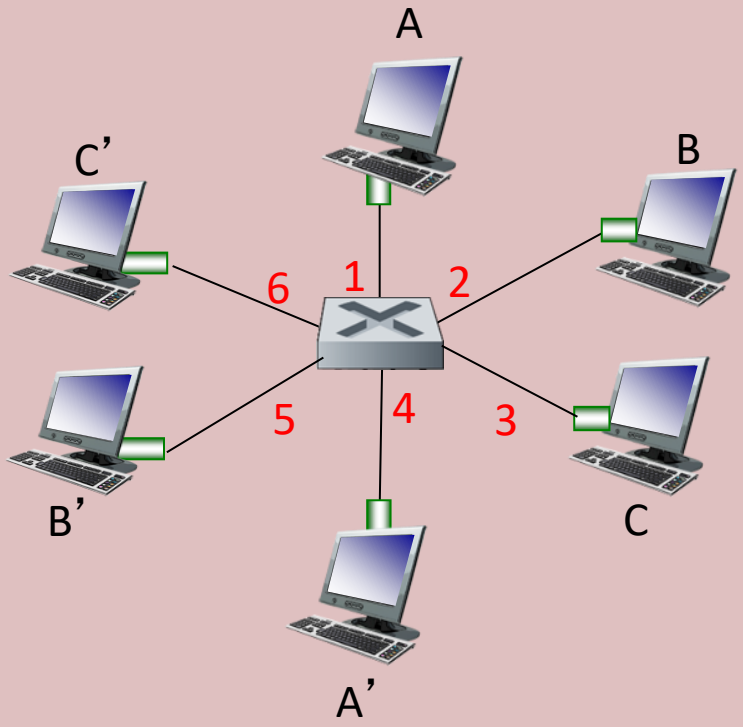
- How are entries created and maintained in a switch table?
  - Something like a routing protocol?



switch with six interfaces (1,2,3,4,5,6)

Switch: Self-Learning

- Switch *learns* which hosts can be reached through which interfaces
  - When frame received, switch "learns" location of sender: incoming LAN segment
  - Records sender/location pair in switch table



switch with six interfaces (1,2,3,4,5,6)

Switch: Frame Filtering/Forwarding

RECEIVE-FRAME:

Record incoming link, MAC address of sending host

Index switch table using MAC destination address

if entry found for destination then {

    if destination on segment from which frame arrived

        then drop frame (why?)

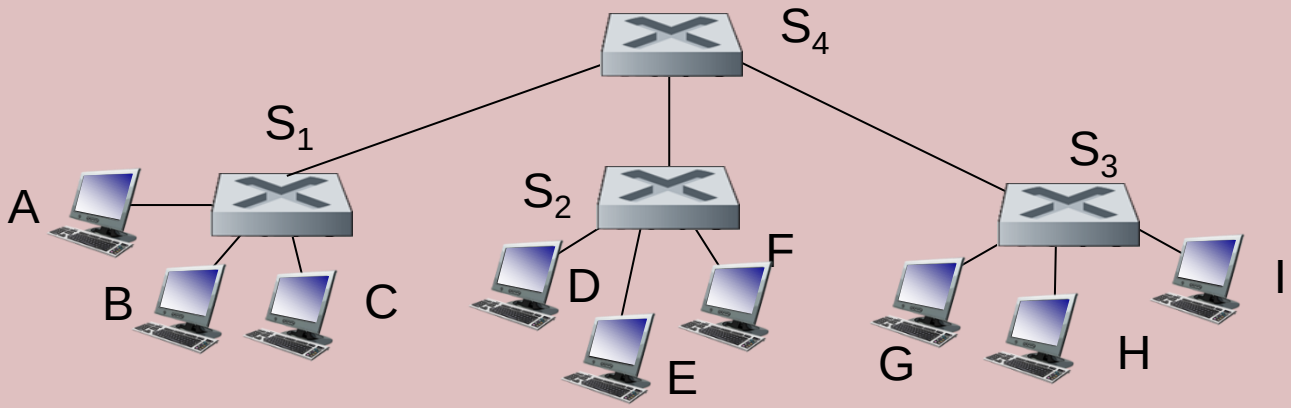
    else forward frame on interface indicated by entry

    }

else flood *#forward on all interfaces except arriving interface*

Interconnecting Switches

Self learning switches can be connected together:



- Q: Sending from *A* to *G* – how does S1 know to forward frame destined to G via S4 and S3?
  - A: Self learning! (works exactly the same as in single-switch case!)

Switches vs. Routers

Both are store-and-forward

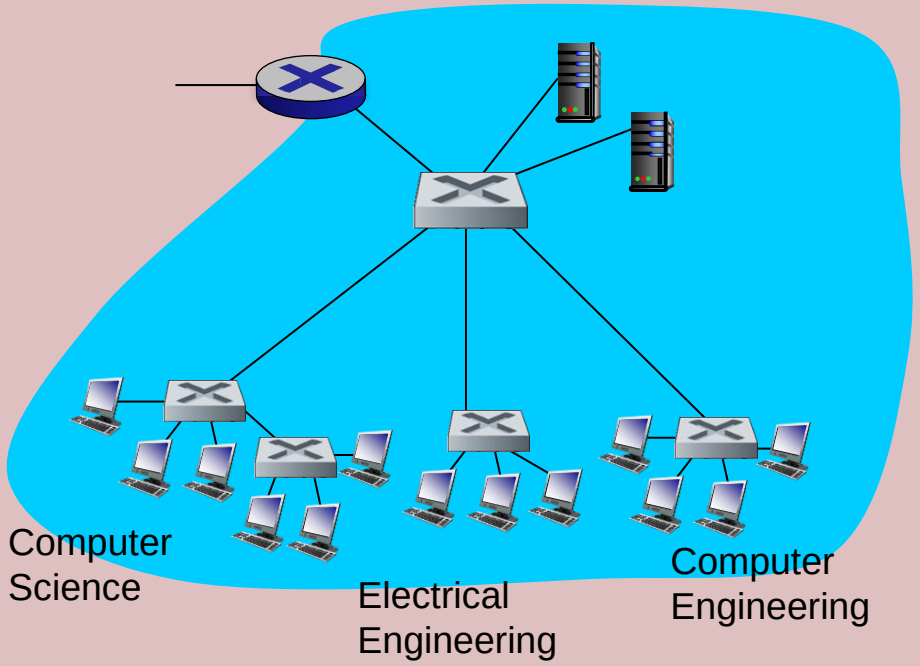
- *Routers:* network-layer devices (examine network-layer headers)
- *Switches:* link-layer devices (examine link-layer headers)

- Both have forwarding tables:
- *Routers:* compute tables using routing algorithms, IP addresses
- *Switches:* learn forwarding table using flooding, learning, MAC addresses

Virtual LAN (VLAN)

Scenario

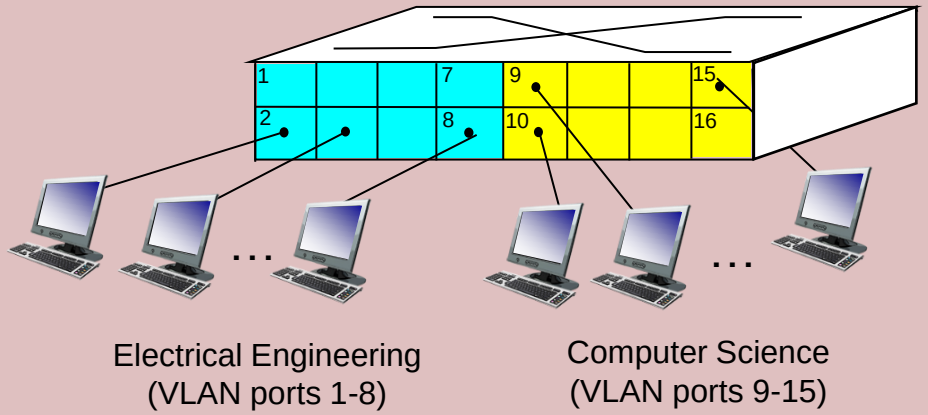
- CS user moves office to EE, but wants to connect to CS switch
- Single broadcast domain:
  - All layer-2 broadcast traffic (ARP, DHCP, unknown location of destination MAC address) must cross entire LAN
  - Security/privacy, efficiency issues

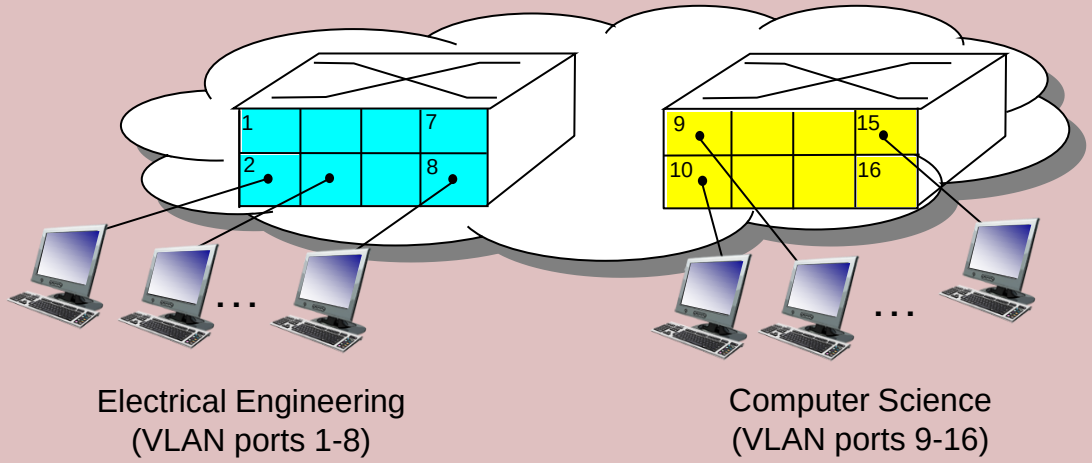


VLAN

Virtual Local Area Network

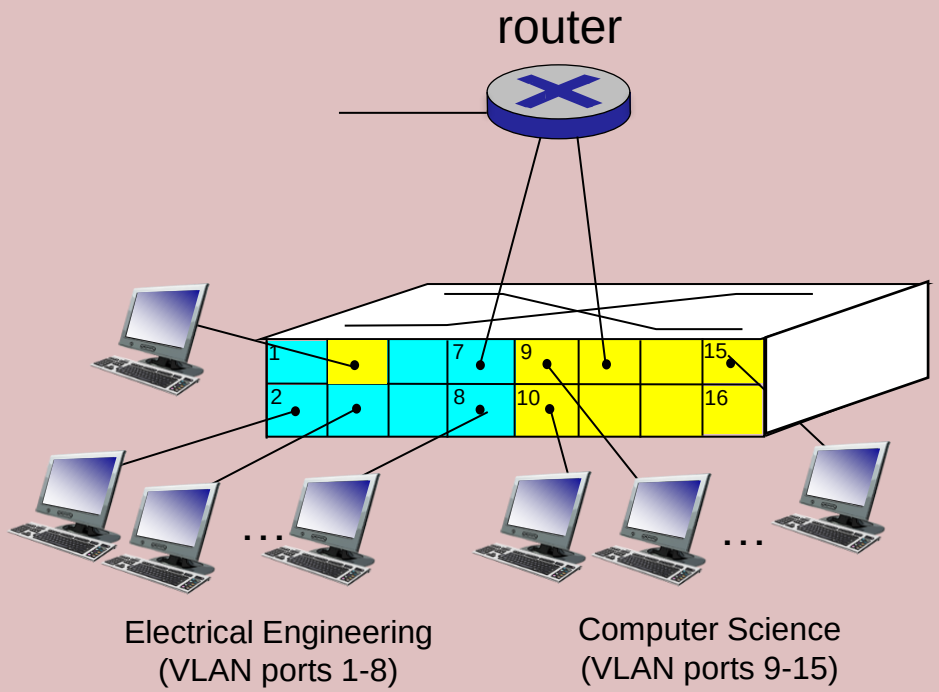
- Switch(es) supporting VLAN capabilities can be configured to define multiple *virtual* LANs over single physical LAN infrastructure
- Port-based VLAN: switch ports are grouped (by switch management software) so that a *single* physical switch operates as multiple virtual switches.



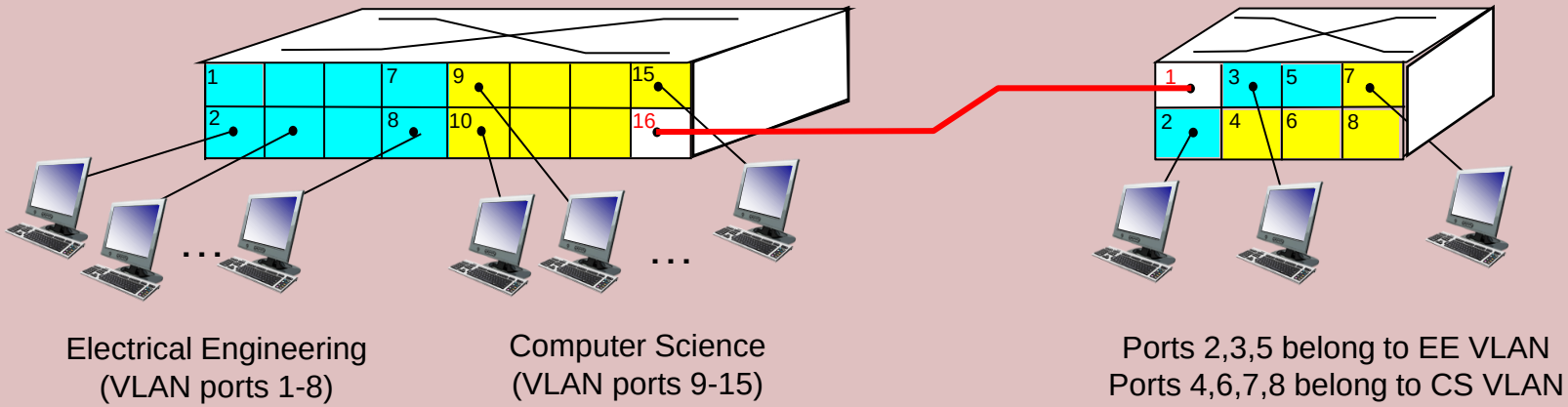


Port-Based VLAN

- *Traffic isolation:* frames to/from ports 1-2, 4-8 can *only* reach ports 1-2, 4-8
  - Can also define VLAN based on MAC addresses of endpoints, rather than switch port
- Dynamic membership: ports can be dynamically assigned among VLANs
- Forwarding between VLANs: done via routing (just as with separate switches)
  - In practice, vendors sell combined switches plus routers



Multi-Switch VLAN



- *Trunk port:* carries frames between VLANs defined over multiple physical switches

Thank You!