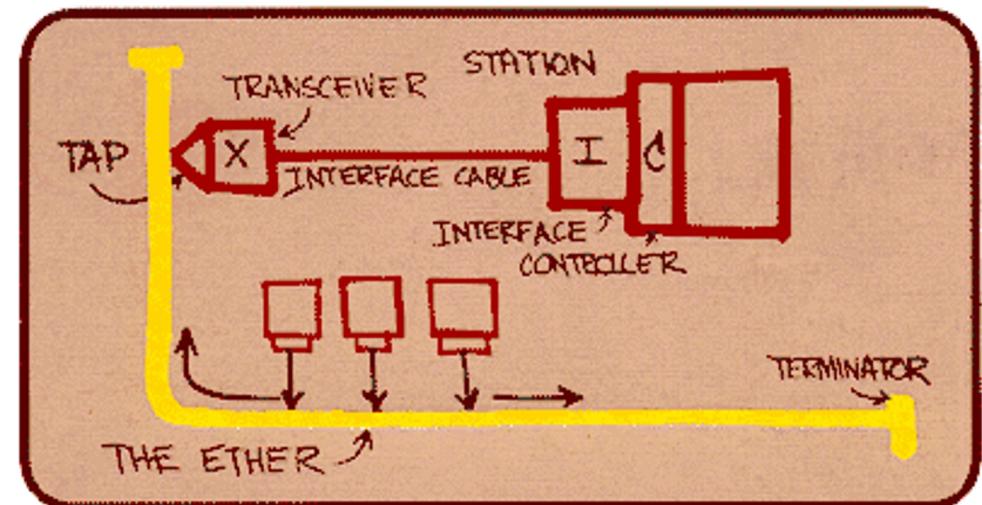




Link Layer/Physical Layer: Types of Links, MAC addresses, ARP, Ethernet, Multiple Access, Switching

Michael Kirkedal Thomsen

Based on slides compiled by Marcos Vaz Salles



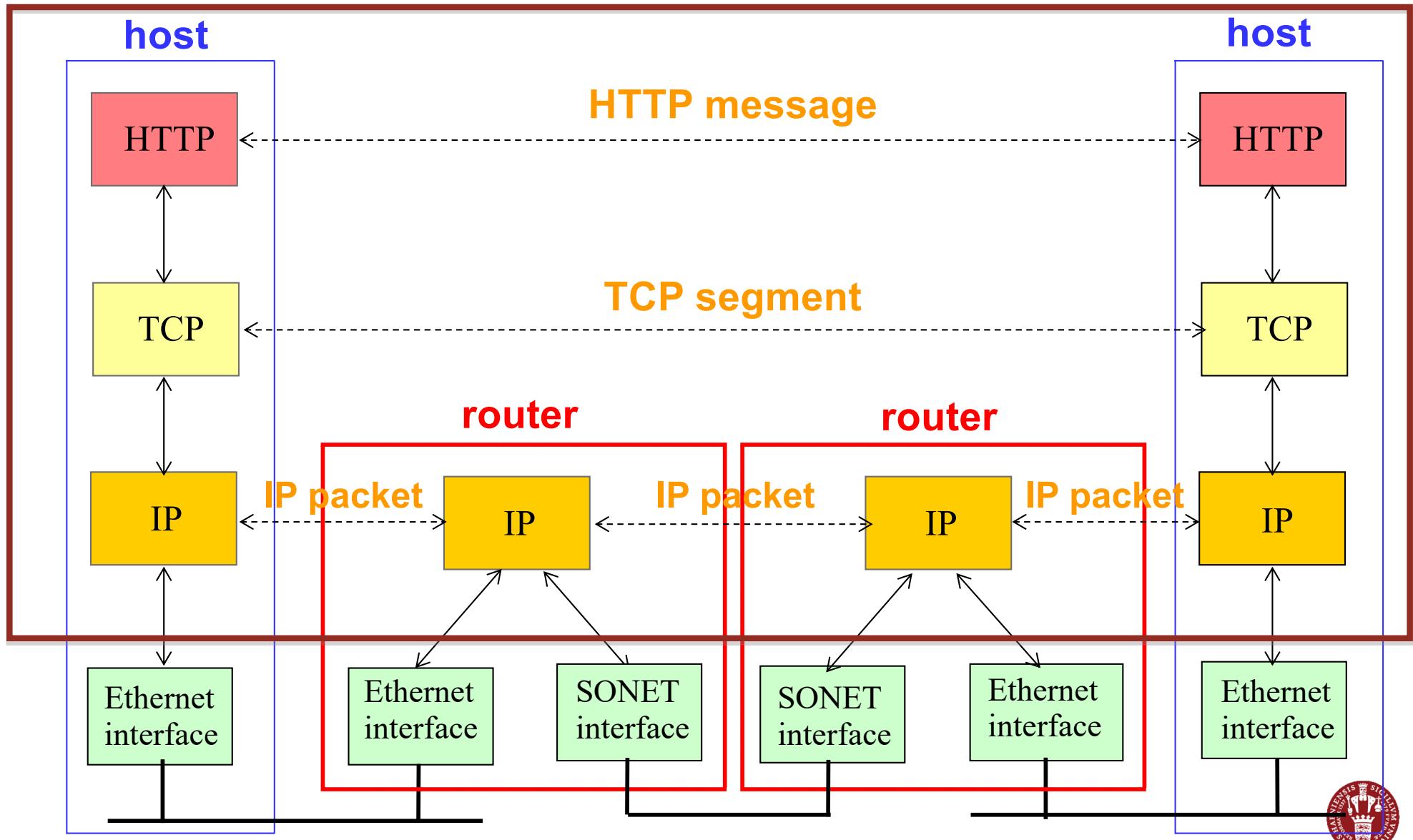
What should we learn today?



- Link layer
- Identify the main functions of the link layer: framing, error detection, and multiple access
- Define the following methods for multiple access: channel partitioning, taking turns, random access
- Describe what MAC addresses are as well as the Address Resolution Protocol (ARP)
- Explain the multiple access mechanisms of Ethernet and their rationale
- Define hubs, switches, and routers
- Describe the self-learning mechanism used in switches to build forwarding tables
- Discuss network configurations consisting of subnets and LAN segments



Recap: Our course so far...

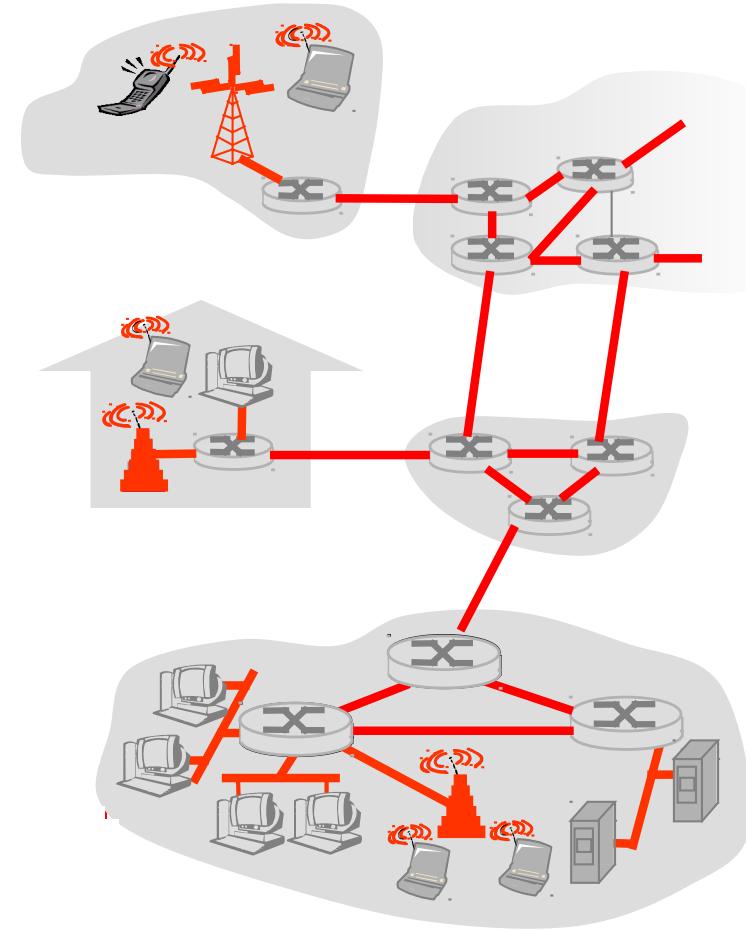


Link Layer: Introduction

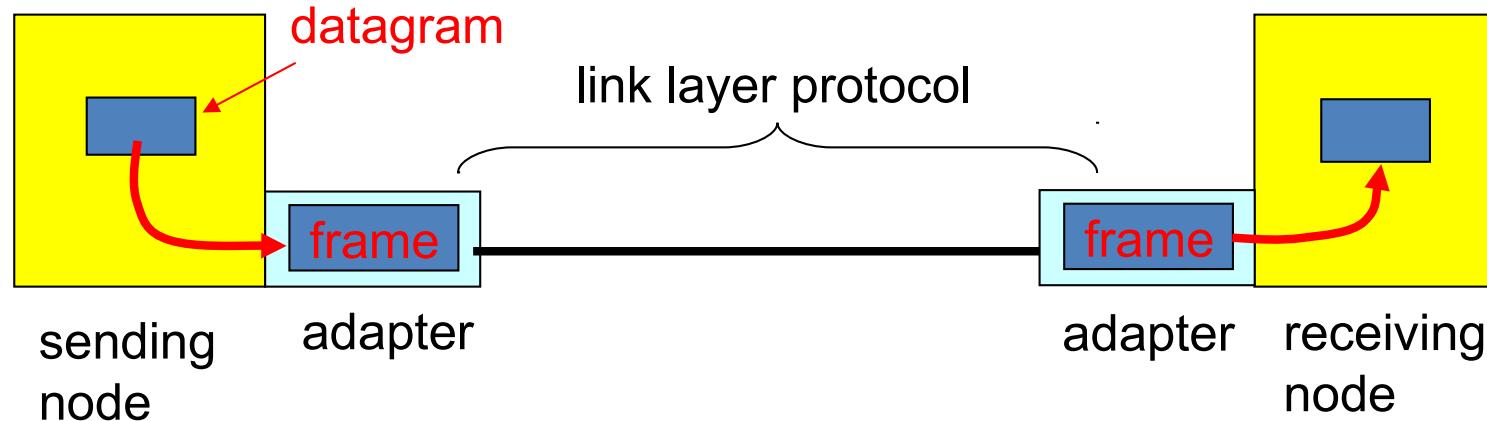
Terminology:

- hosts and routers are **nodes**
- communication channels that connect adjacent nodes along communication path are **links**
 - Wired links
 - Wireless links
 - LANs
- layer-2 packet is a **frame**, encapsulates datagram

data-link layer has responsibility of transferring datagram from one node to **physically adjacent** node over a link



Digital adaptors Communicating



- Link layer implemented in adaptor (network interface card)
 - Ethernet card, PCMCIA card, 802.11 card
- Sending side:
 - Encapsulates datagram in a frame
 - Adds error checking bits, flow control, etc.
- Receiving side
 - Looks for errors, flow control, etc.
 - Extracts datagram and passes to receiving node



Framing

- Break sequence of bits into a frame
 - Typically implemented by the network adaptor
- Sentinel-based
 - Delineate frame with special pattern (e.g., 01111110)



- Problem: what if special patterns occurs within frame?
- Solution: escaping the special characters
 - E.g., sender always inserts a 0 after five 1s
 - ... and receiver always removes a 0 appearing after five 1s
- Similar to escaping special characters in C programs



Framing (Continued)

- Counter-based
 - Include the payload length in the header
 - ... instead of putting a sentinel at the end
 - Problem: what if the count field gets corrupted?
 - Causes receiver to think the frame ends at a different place
 - Solution: catch later when doing error detection
 - And wait for the next sentinel for the start of a new frame
- Clock-based
 - Make each frame a fixed size
 - No ambiguity about start and end of frame
 - But, may be wasteful



Error Detection

- Errors are unavoidable
 - Electrical interference, thermal noise, etc.
- Error detection
 - Transmit extra (redundant) information
 - Use redundant information to detect errors
 - Extreme case: send two copies of the data
 - Trade-off: accuracy vs. overhead



Error Detection Techniques

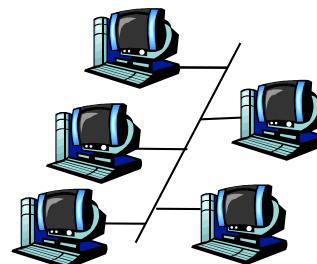
- Parity check
 - Add an extra bit to a 7-bit code
 - Odd parity: ensure an odd number of 1s
 - E.g., 0101011 becomes 01010111
 - Even parity: ensure an even number of 1s
 - E.g., 0101011 becomes 01010110
- Checksum
 - Treat data as a sequence of 16-bit words
 - Compute a sum of all 16-bit words, with carries and wraparounds
 - Transmit the sum (1s complement of this) along with the packet
- Cyclic Redundancy Check (CRC)
 - See book



Multiple Access Links and Protocols

Two types of “links”:

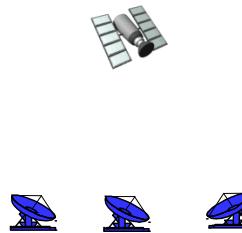
- point-to-point
 - PPPfordial-upaccess
 - point-to-pointlinkbetweenEthernetswitchandhost
- broadcast (shared wire or medium)
 - old-fashionedEthernet
 - upstreamHFC
 - 802.11wirelessLAN



shared wire (e.g.,
cabled Ethernet)



shared RF
(e.g., 802.11 WiFi)



shared RF
(satellite)



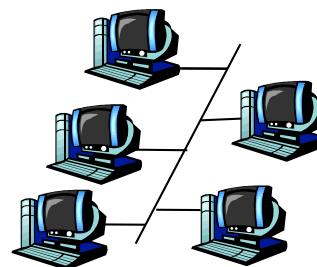
humans at a
cocktail party
(shared air, acoustical)

Source: Kurose & Ross



Multiple Access Links and Protocols

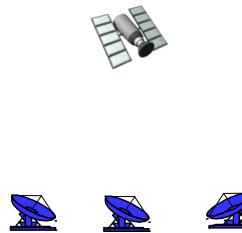
- Single shared broadcast channel
 - Avoid having multiple nodes speaking at once
 - Otherwise, collisions lead to garbled data
- Multiple access protocol
 - Distributed algorithm for sharing the channel
 - Algorithm determines which node can transmit
- How would you design such a protocol?



shared wire (e.g.,
cabled Ethernet)



shared RF
(e.g., 802.11 WiFi)



shared RF
(satellite)



humans at a
cocktail party
(shared air, acoustical)

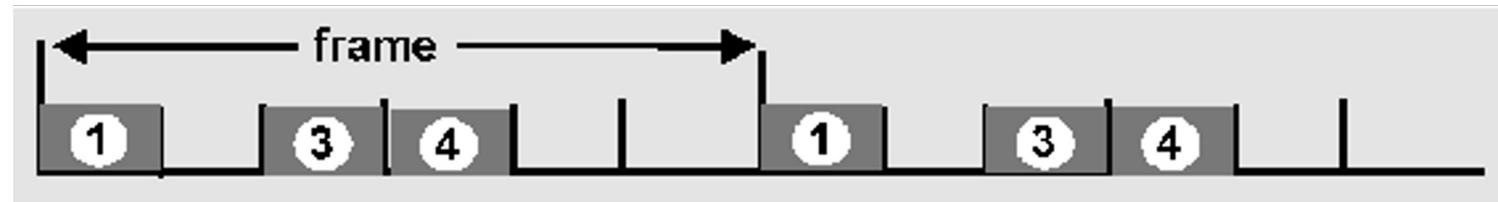
Source: Kurose & Ross



Channel Partitioning: TDMA

TDMA: time division multiple access

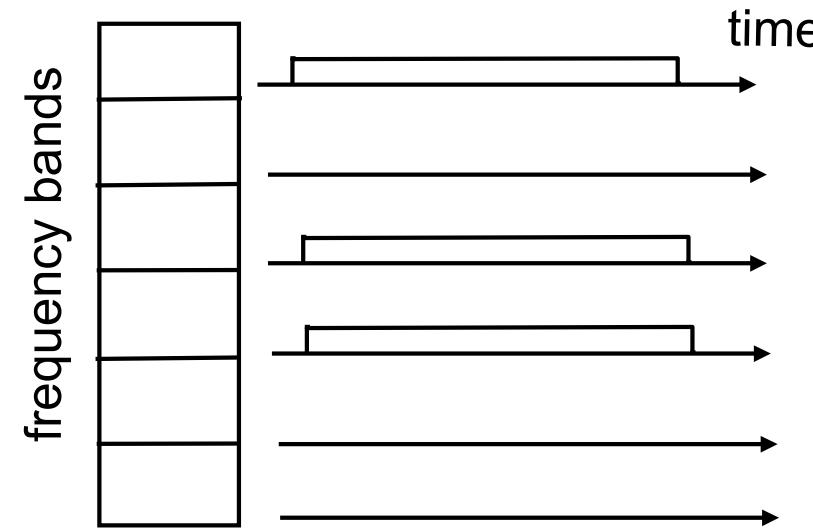
- Access to channel in "rounds"
 - Each station gets fixed length slot in each round
- Time-slot length is packet transmission time
 - Unused slots go idle
- Example: 6-station LAN with slots 1, 3, and 4



Channel Partitioning: FDMA

FDMA: frequency division multiple access

- Channel spectrum divided into frequency bands
 - Each station has fixed frequency band (Wifi channels 1-11)
- Unused transmission time in bands go idle
- Example: 6-station LAN with bands 1, 3, and 4



Source: Freedman



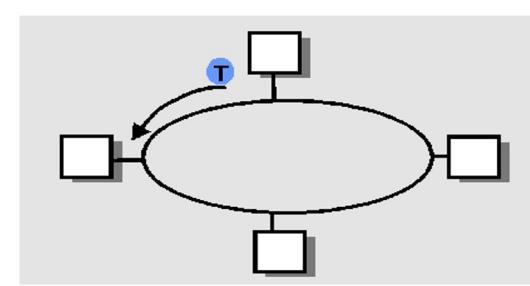
“Taking Turns” MAC protocols

Polling

- Primary node “invites” secondary nodes to transmit in turn
- Concerns:
 - Polling overhead
 - Latency
 - Single point of failure (primary)

Token passing

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



Source:
Freedman



Random Access Protocols

- When node has packet to send
 - Transmit at full channel data rate R .
 - No *a priori* coordination among nodes
- Two or more transmitting nodes → “collision”
- Random access MAC protocol specifies:
 - How to detect collisions
 - How to recover from collisions



Key Ideas of Random Access

- Carrier Sense (CS)
 - *Listen before speaking, and don't interrupt*
 - Checking if someone else is already sending data
 - ... and waiting till the other node is done
- Collision Detection (CD)
 - *If someone else starts talking at the same time, stop*
 - Realizing when two nodes are transmitting at once
 - ...by detecting that the data on the wire is garbled
- Randomness
 - *Don't start talking again right away*
 - Waiting for a random time before trying again



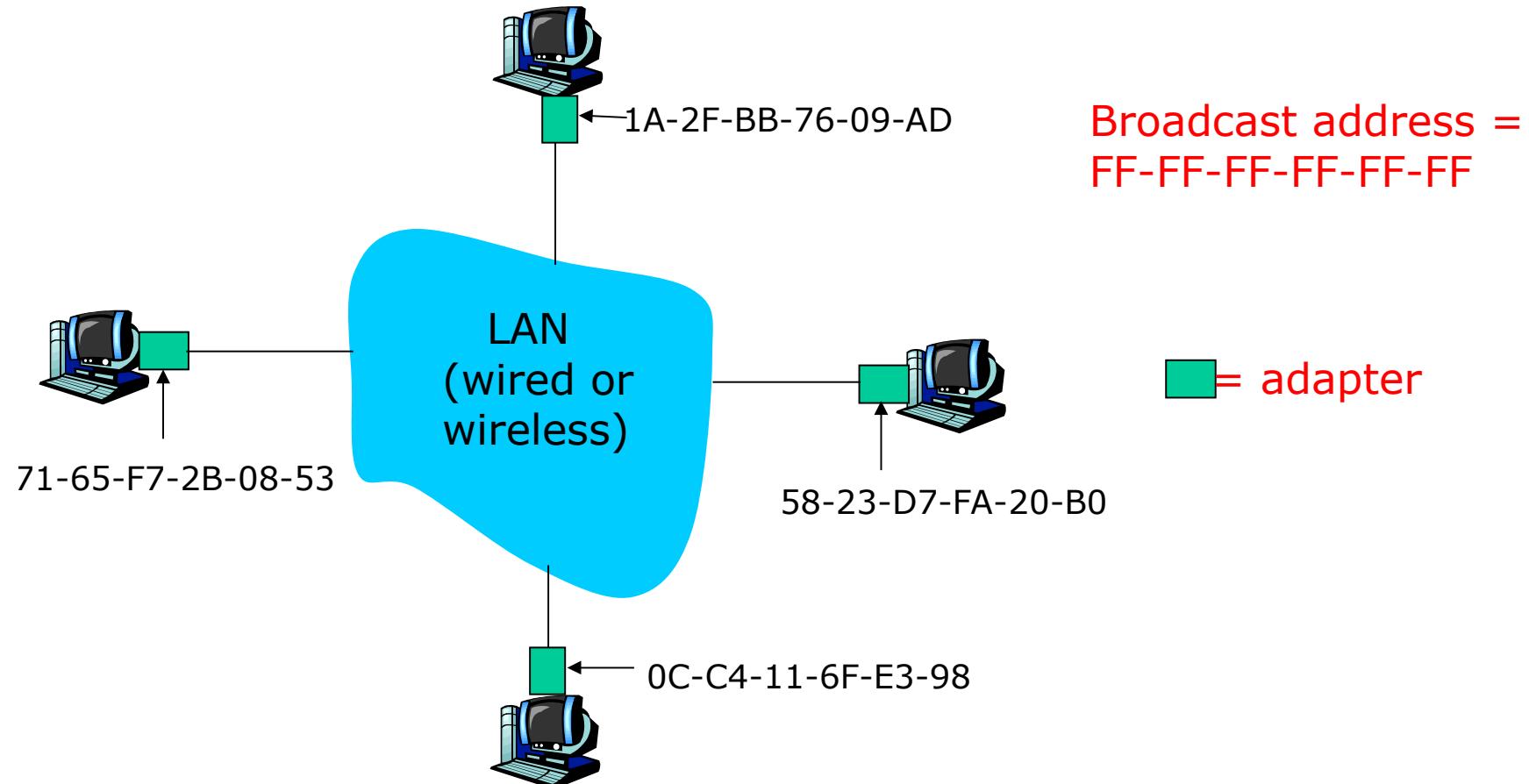
MAC Addresses and ARP

- 32-bit IP address:
 - *network-layer* address
 - used to get datagram to destination IP subnet
- MAC (or LAN or physical or Ethernet) address:
 - function: *get frame from one interface to another physically-connected interface (same network)*
 - 48 bit MAC address (for most LANs)
 - burned in NIC ROM, also sometimes software settable



LAN Addresses and ARP

Each adapter on LAN has unique LAN address



LAN Address (more)

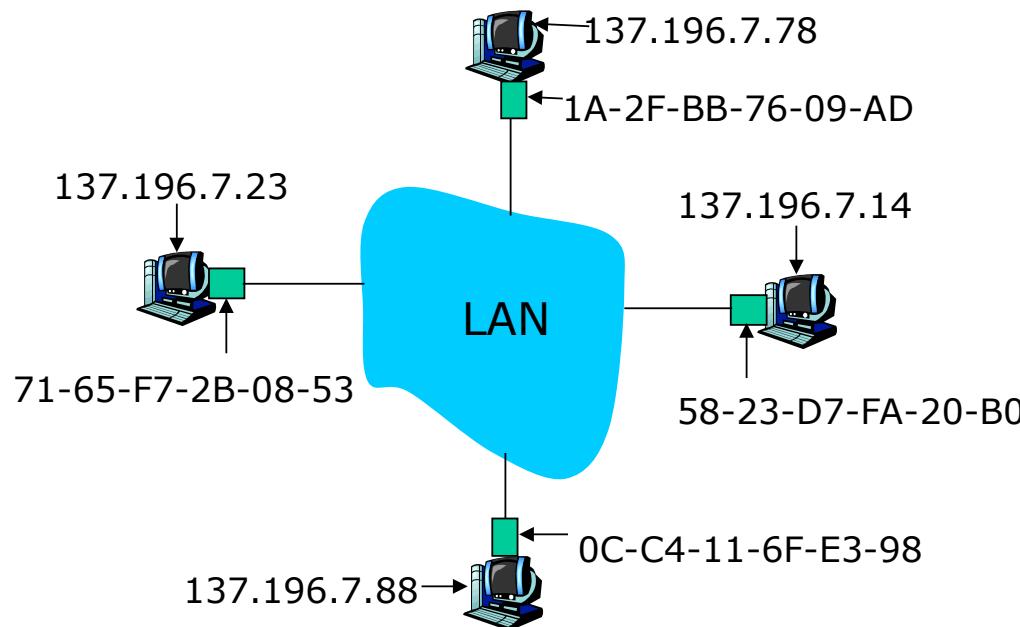
- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space (to assure uniqueness)
- Analogy:
 - MAC address: like Social Security Number
 - IP address: like postal address
- MAC flat address → portability
 - can move LAN card from one LAN to another
- IP hierarchical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



ARP: Address Resolution Protocol

Question: how to determine MAC address of B knowing B's IP address?

- Each IP node (host, router) on LAN has **ARP** table
- ARP 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: Same LAN (network)

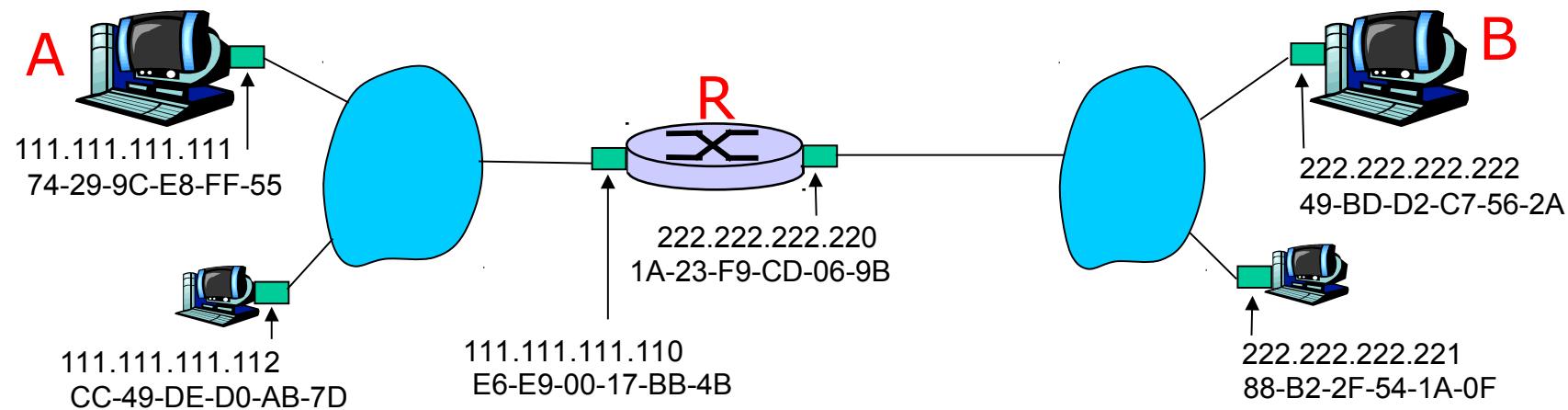
- A wants to send datagram to B, and B's MAC address not in A's ARP table.
- A **broadcasts** ARP query packet, containing B's IP address
 - dest MAC address = FF-FF-FF-FF-FF-FF
 - all machines on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
 - frame sent to A's MAC address (unicast)
- A **caches** (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
 - **soft state:** information that times out (goes away) unless refreshed
- ARP is “plug-and-play”:
 - nodes create their ARP tables *without intervention from net administrator*



Addressing: routing to another LAN

walkthrough: **send datagram from A to B via R.**

- focus on addressing - at both IP (datagram) and MAC layer (frame)
- assume A knows B's IP address
- assume A knows IP address of first hop router, R (how?)
- assume A knows MAC address of first hop router interface (how?)
- assume R knows B's MAC address (how?)

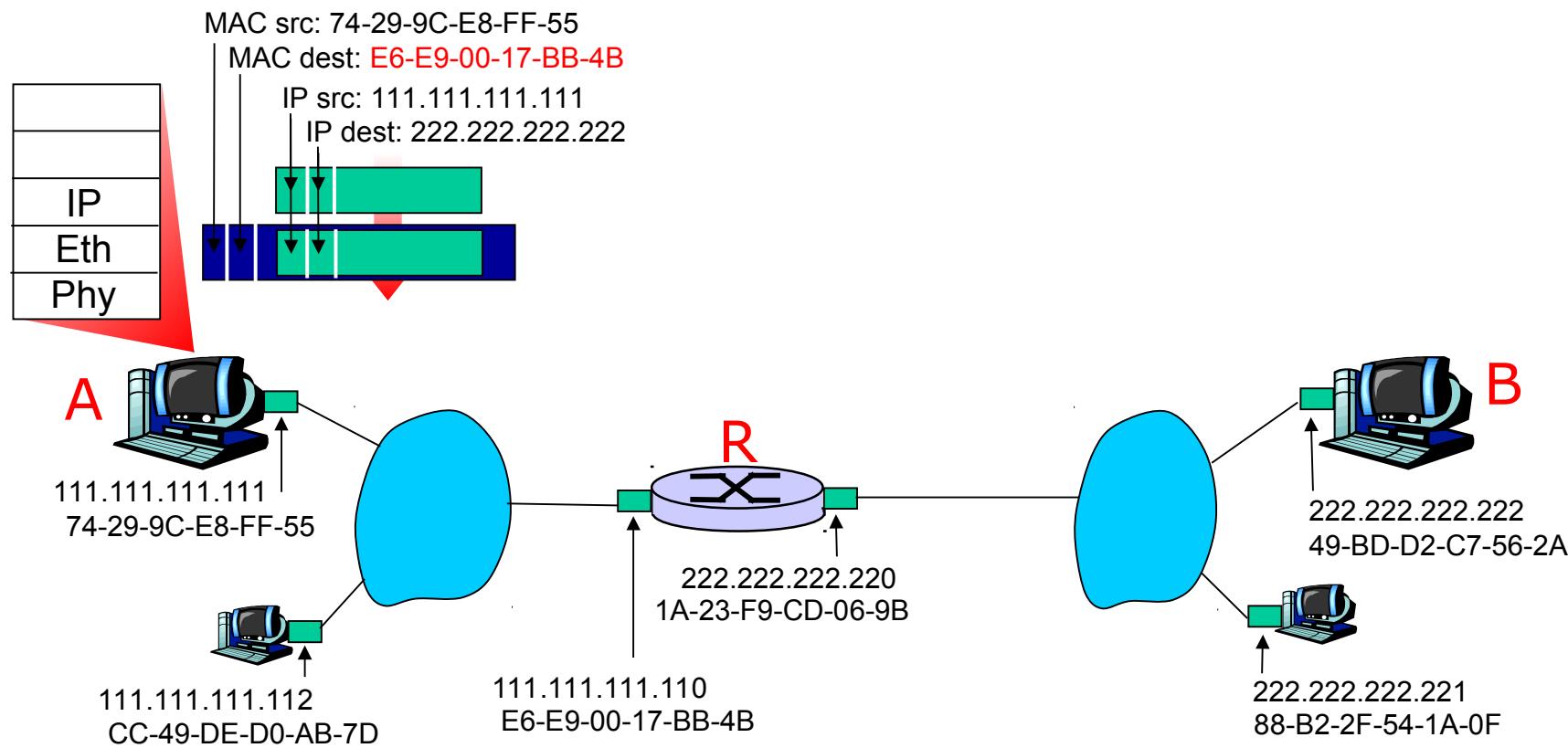


Source: Kurose & Ross (partial)



Addressing: routing to another LAN

- A creates IP datagram with IP source A, destination B
- A creates link-layer frame with R's MAC address as dest, frame contains A-to-B IP datagram

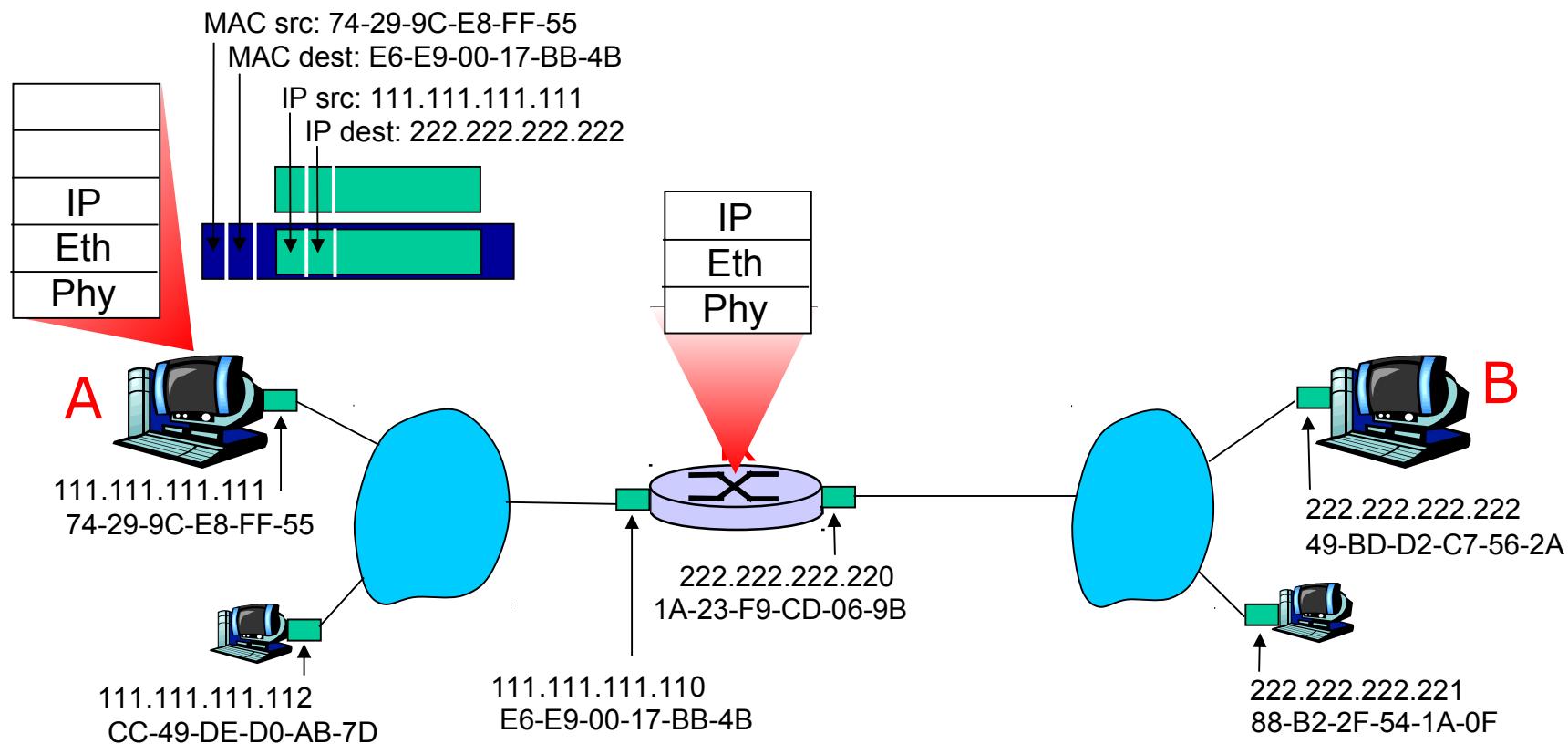


Source: Kurose & Ross (partial)



Addressing: routing to another LAN

- frame sent from A to R
- frame received at R, datagram removed, passed up to IP

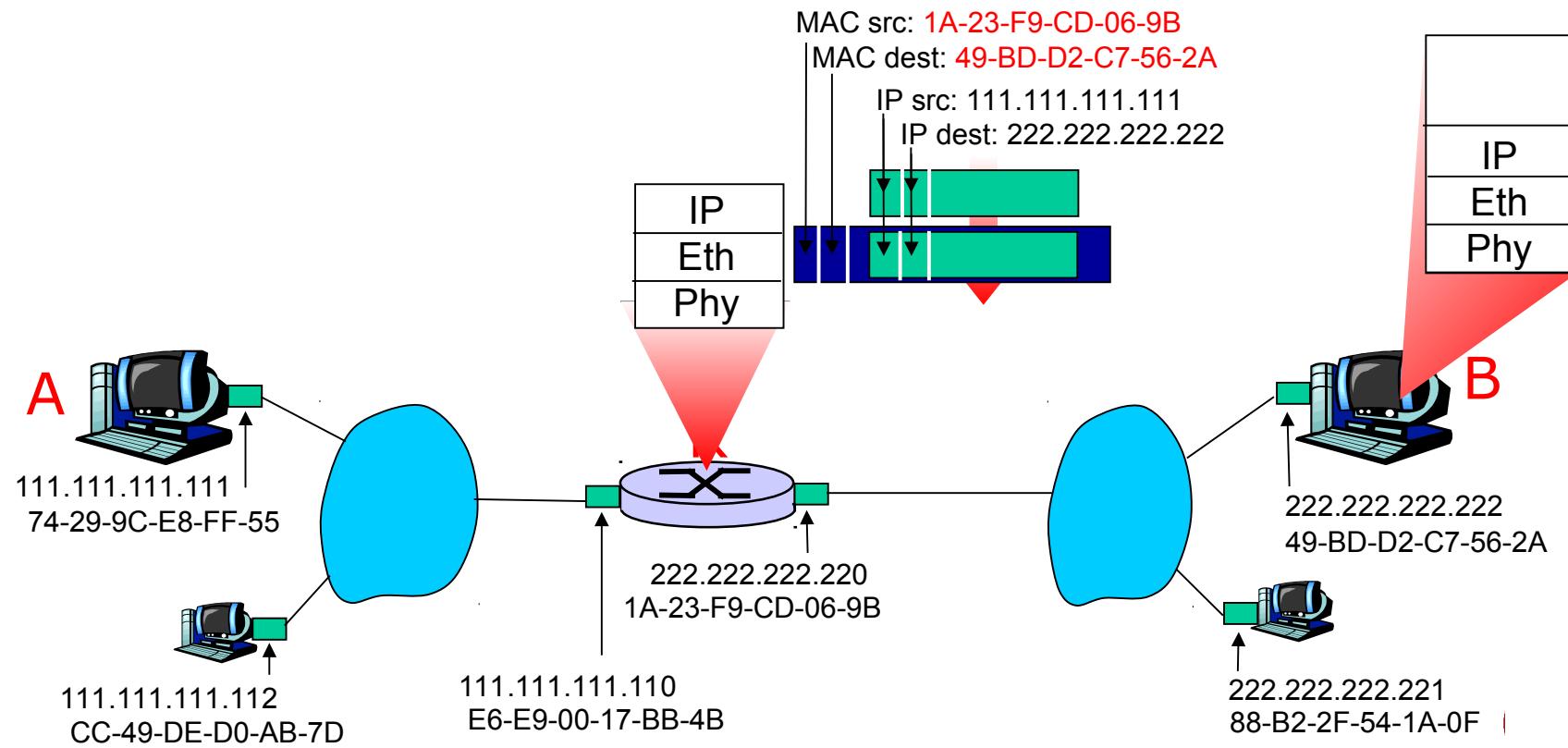


Source: Kurose & Ross (partial)



Addressing: routing to another LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram

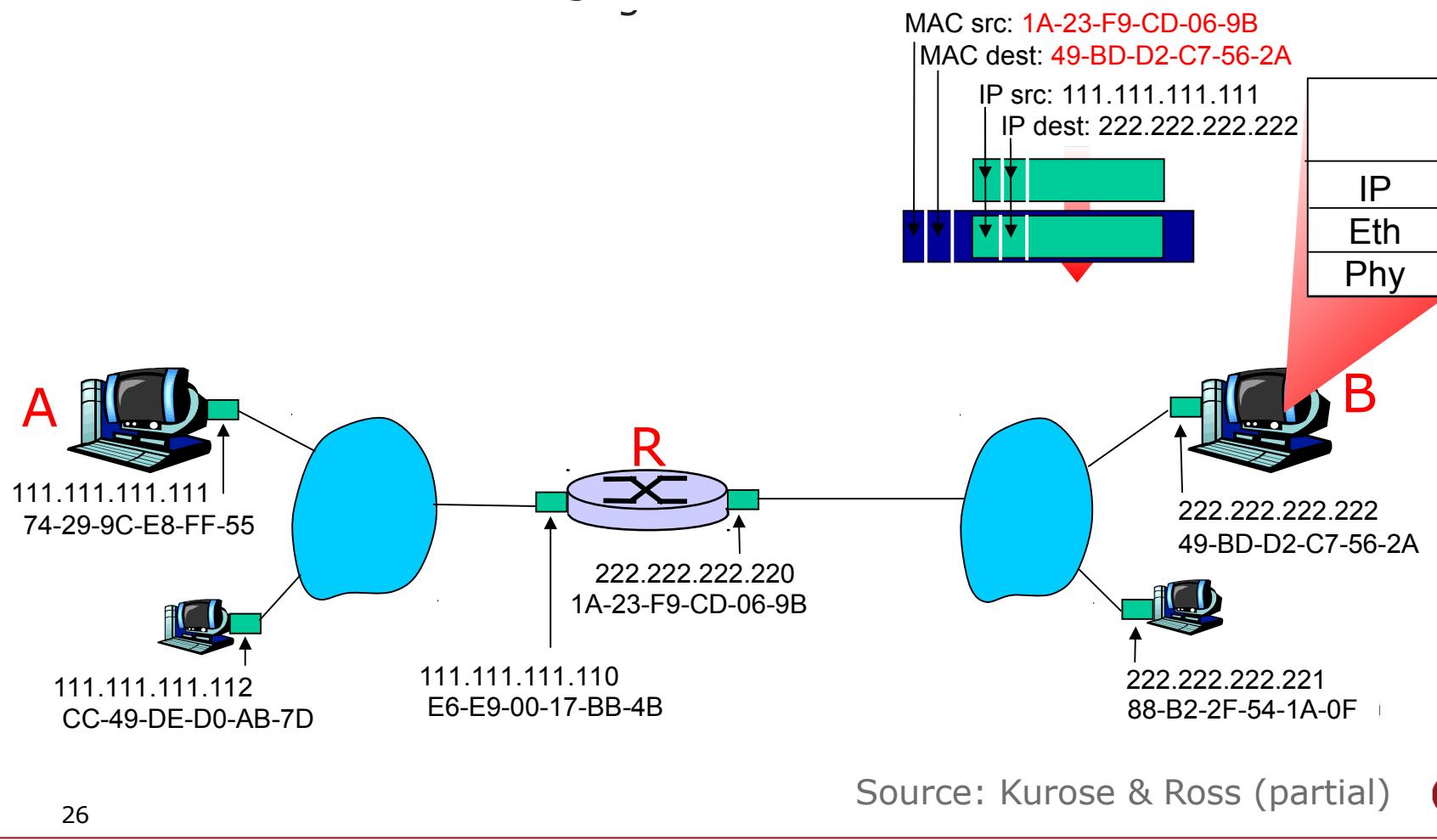


Source: Kurose & Ross (partial)



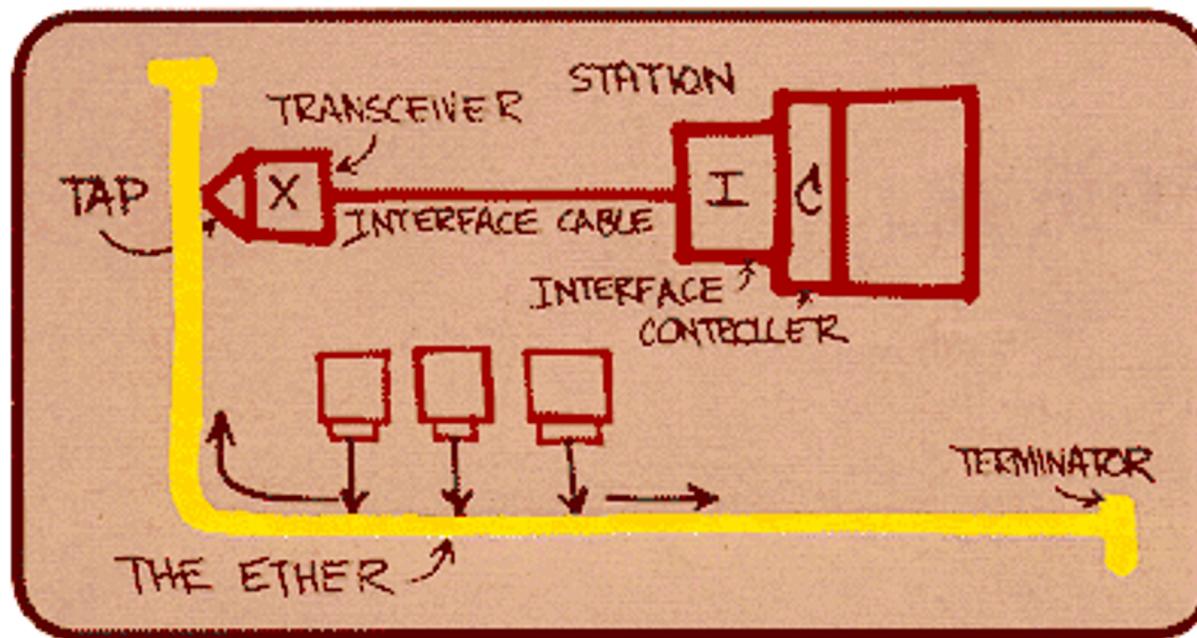
Addressing: routing to another LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



Ethernet

- Dominant wired LAN technology, first widely used
- Simpler, cheaper than token LANs and ATM
- Kept up with speed race: 10 Mbps – 10 Gbps



Metcalfe's
Ethernet
sketch

Source: Freedman

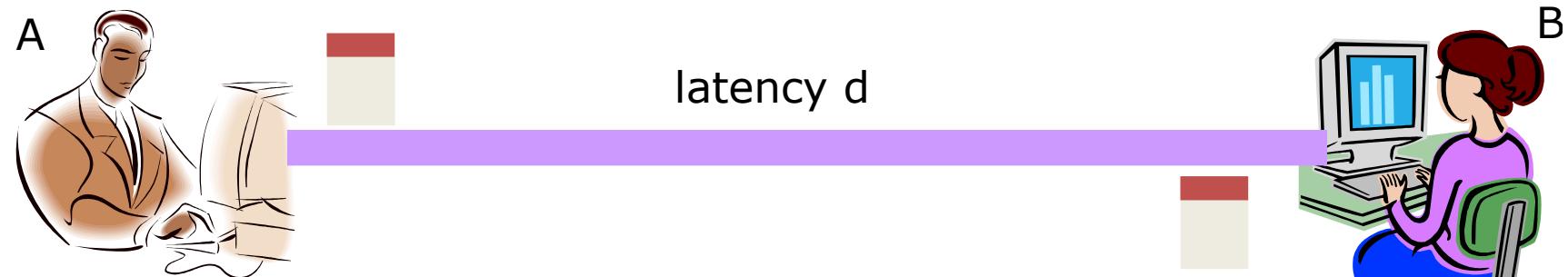


Ethernet Uses CSMA/CD

- **Carrier Sense (CS)**
 - *Listen before speaking, and don't interrupt*
 - wait for link to be idle before transmit
- **Collision Detection (CD)**
 - *If someone else starts talking at the same time, stop*
 - listen while transmitting
 - No collision: transmission complete
 - Collision: abort and send jam signal
- **Randomness (exponential back-off)**
 - *Don't start talking again right away*
 - After collision, wait a random time before retry
 - After m^{th} collision, choose K randomly from $\{0, \dots, 2^m - 1\}$
 - ... and wait for $K * 64$ byte times before retry



Limitations on Ethernet Length

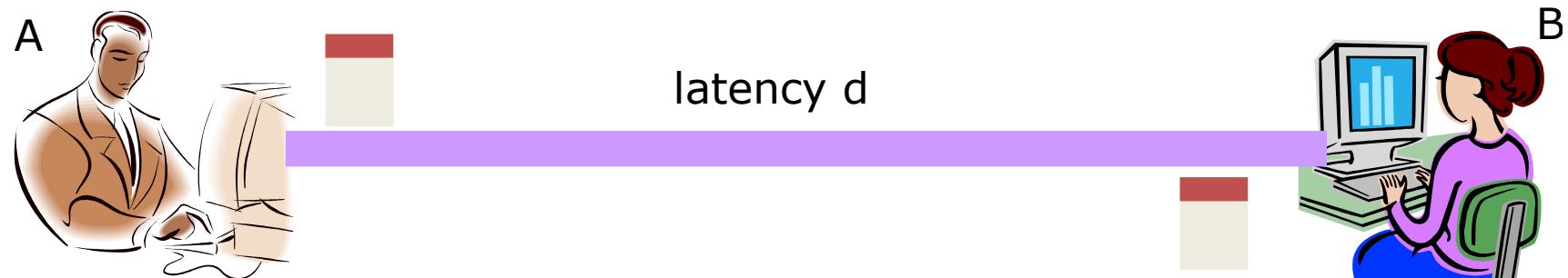


- Latency depends on physical length of link
 - Time to propagate a packet from one end to the other
- Suppose A sends a packet at time t
 - And B sees an idle line just before time $t+d$, so transmits
- B detects a collision, and sends jamming signal
 - But A doesn't see collision till $t+2d$

Source: Freedman



Limitations on Ethernet Length



- A needs to wait for time $2d$ to detect collision
 - So, A should keep transmitting during this period
 - ... and keep an eye out for a possible collision
- Imposes restrictions on Ethernet
 - Max length of wire: 2500 meters
 - Min length of packet: 512 bits (64 bytes)

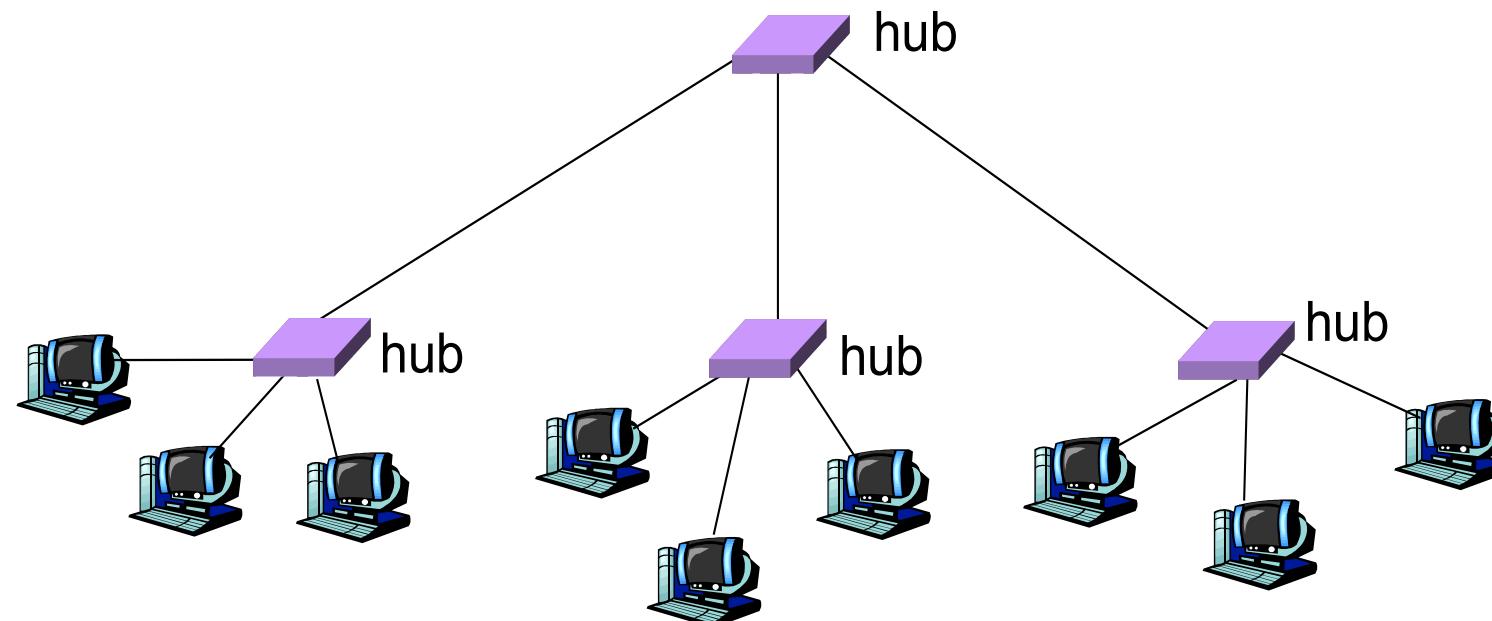
Source: Freedman



Physical Layer: Hubs

... physical-layer (“dumb”) repeaters:

- bits coming in one link go out *all* other links at same rate
- all nodes connected to hub can collide with one another
- no frame buffering
- no CSMA/CD at hub: host NICs detect collisions



Source: Kurose & Ross and Freedman (partial)



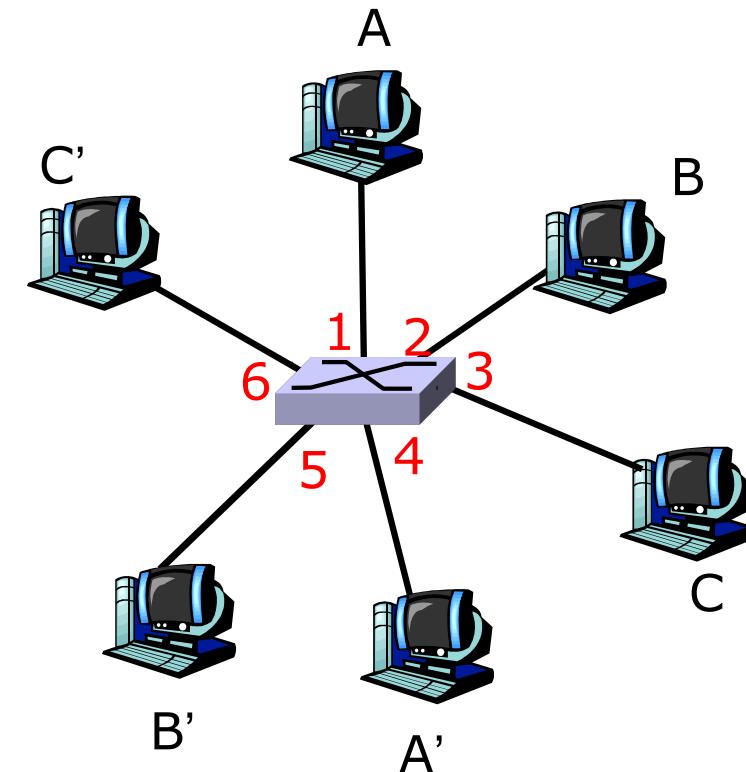
Limitations of Repeaters and Hubs

- One large shared link
 - Each bit sent everywhere, aggregate throughput limited
- Cannot support multiple LAN technologies
 - Does not buffer or interpret frames
 - So, can't interconnect different rates or formats
- Limitations on maximum nodes and distances



Link Layer: Switches

- Connects two or more LAN segments at the link layer
 - Extracts destination address from the frame
 - Looks up the destination in a table, forwards to appropriate
- Each segment can carry its own traffic
 - Concurrent traffic between LANs/host: A to B while A' to B'

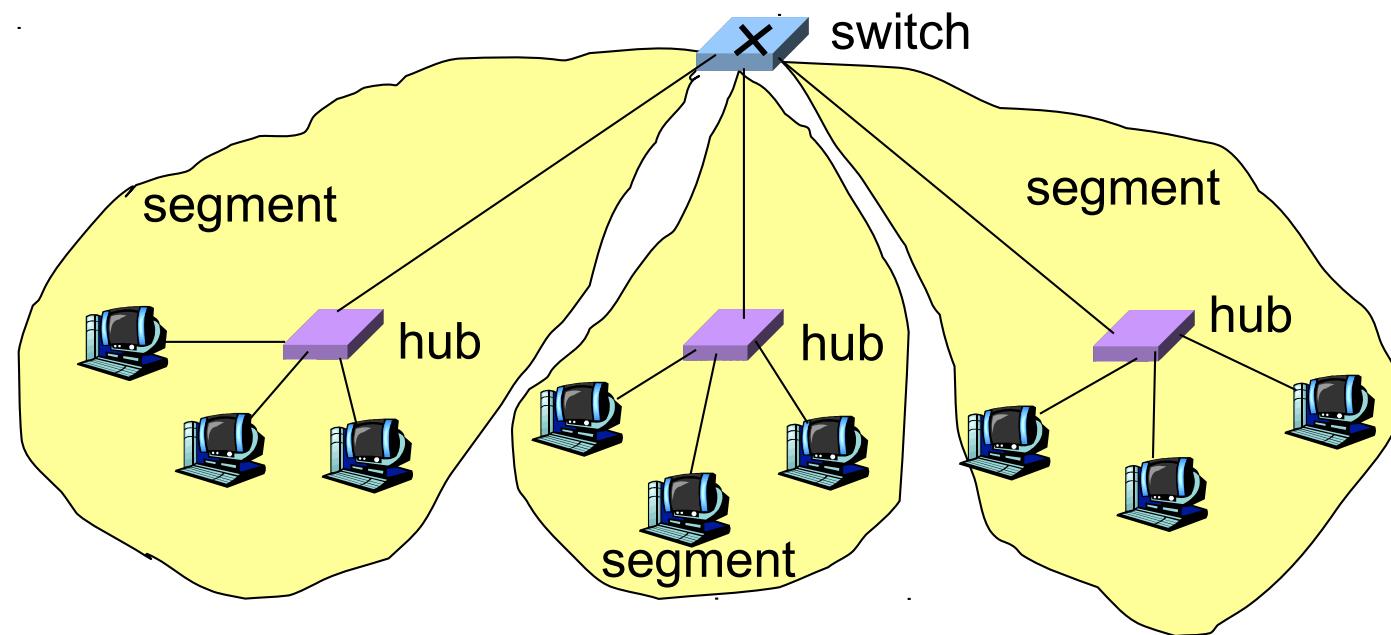


switch with six interfaces



Switches: Traffic Isolation

- Switch breaks subnet into LAN segments
- Switch filters packets
 - Frame only forwarded to the necessary segments
 - Segments can support separate transmissions



Source: Freedman



Advantages Over Hubs/Repeaters

- Only forwards frames as needed
 - E.g. to destination segments or for broadcast traffic
 - Reduces unnecessary traffic on segments
- Extends the geographic span of the network
 - Ethernet collisions (and distance limitations) only on segment
- Improves privacy by limiting scope of frames
 - Hosts can only “snoop” the traffic traversing *their* segment
- Can join segments using different technologies
- Allows for simultaneous transmissions



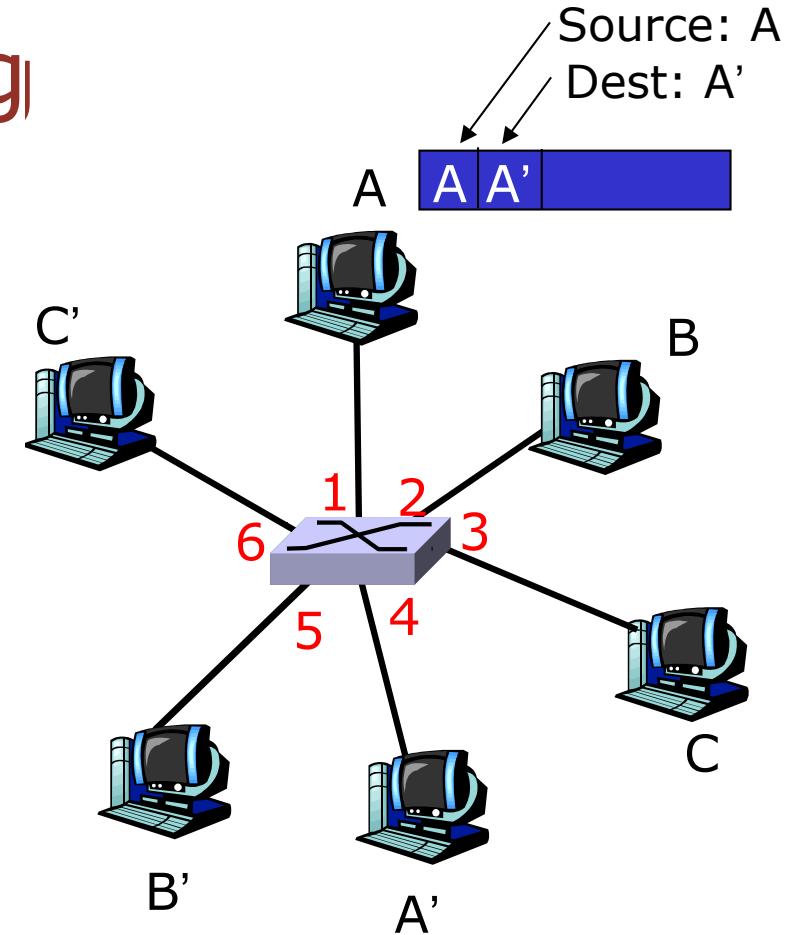
Disadvantages Over Hubs/Repeaters

- Delay in forwarding frames
 - Bridge/switch must receive frame, parse, lookup, and send
 - Storing and forwarding the packet introduces delay
 - Solution: **cut-through switching** (start send after receive header)
- Need to learn where to forward frames
 - Forwarding table: destination MAC → outgoing interface
 - Needs to construct forwarding table, ideally w/o static config
 - Solution: **self-learning**
- Higher cost
 - More complicated devices that cost more money



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



MAC addr	interface	TTL
A	1	60

Switch table
(initially empty)



Switch: frame filtering/forwarding

When frame received:

1. record link associated with sending host
2. index switch table using MAC dest address
3. if entry found for destination
 - then {
 - if dest on segment from which frame arrived
 - then drop the frame
 - else forward the frame on interface indicated

{

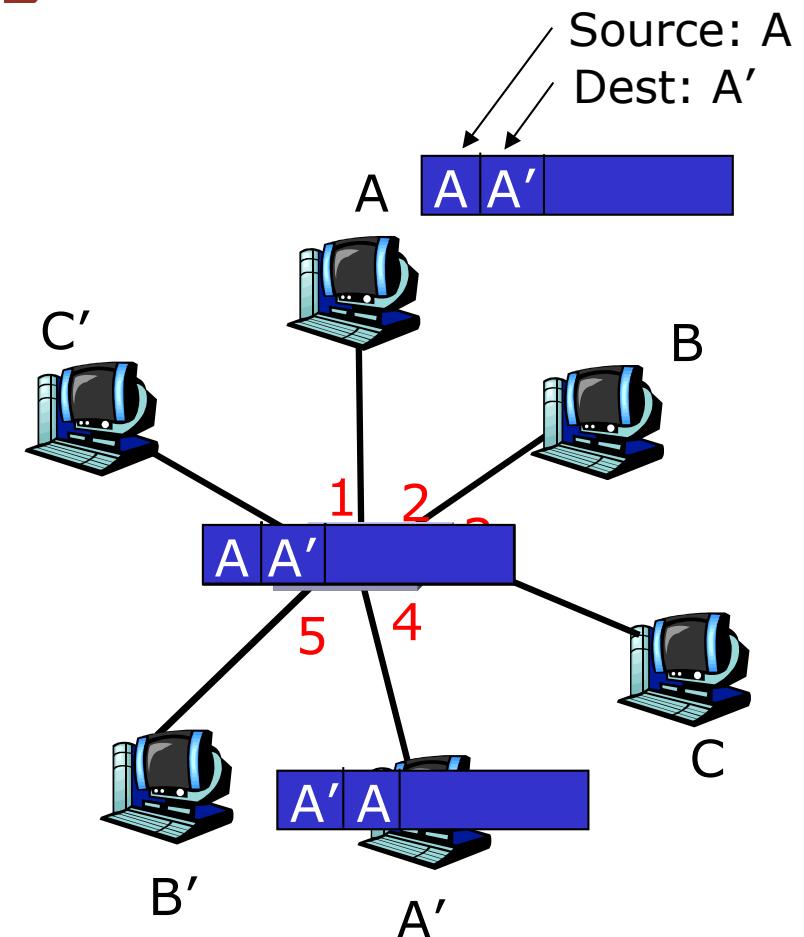
else flood

forward on all but the interface
on which the frame arrived



Switch: self-learning

- frame destination C' unknown: **flood**
- destination A location known: **selective send**



**Switch table
(initially empty)**

MAC addr	interface	TTL
A	1	60
A'	4	60

Source: Freedman





Comparing Hubs, Switches, Routers

	Hub / Repeat er	Bridge / Switch	IP Route r
Traffic isolation	no	yes	yes
Plug and Play	yes	yes	no
Efficient routing	no	no	yes
Cut through	yes	yes	no

Source: Freedman



High-density switching

Source:
Freedman
(partial)



Partial view of “rack”



48-port switch



Facebook rack

- Each rack has 42 U (“pizza boxes”)
- Typically servers + 1-2 “top-of-rack” switch(es)



Summary

- Link Layer
 - Framing, error detection, multiple access
 - Channel partitioning, taking turns, random access
 - MAC addresses and ARP
 - Ethernet, CSMA/CD
 - Hubs, switches, routers
 - Subnets, LAN segments, self-learning in switches

