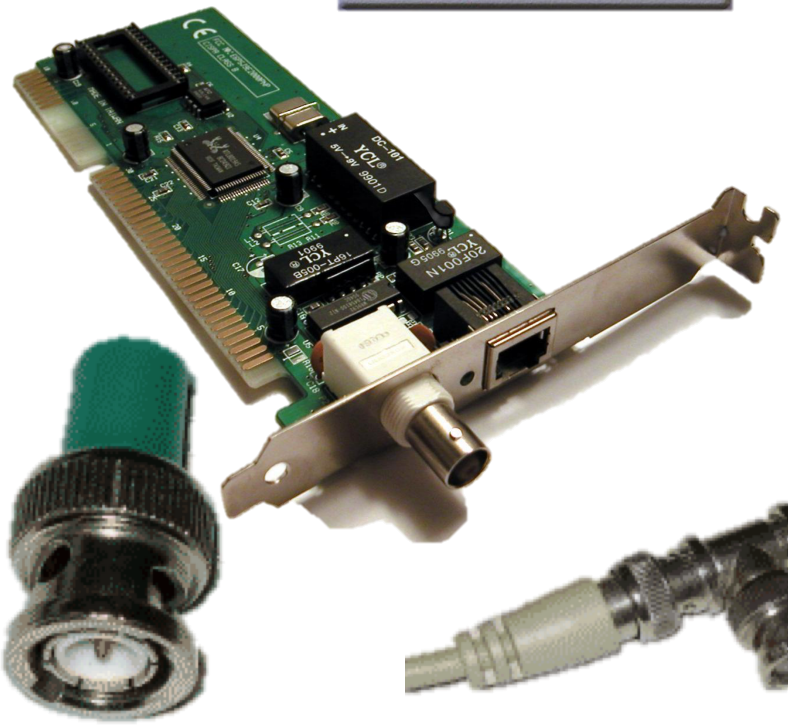
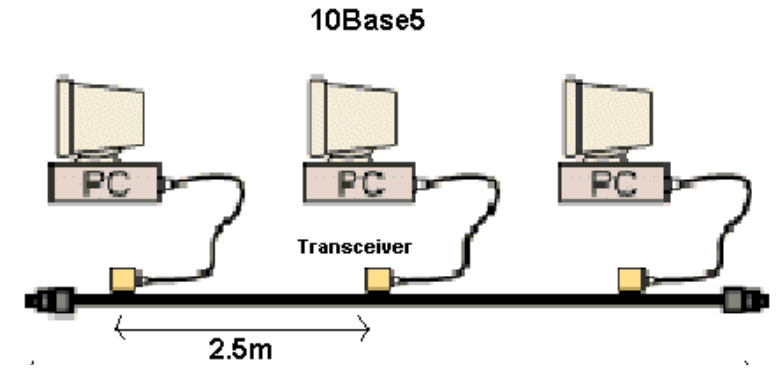
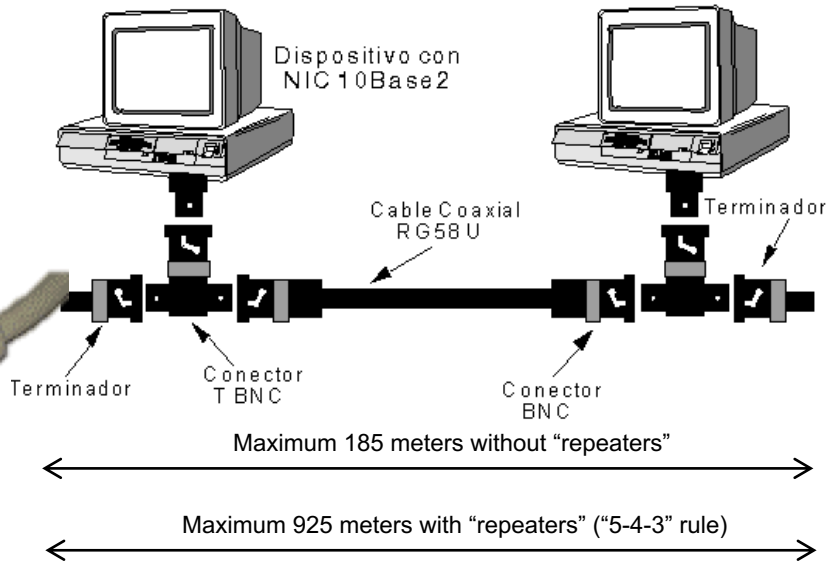
Link Layer (continued)

10Base5 (thick Ethernet) & 10Base2 (thin Ethernet)

FYI

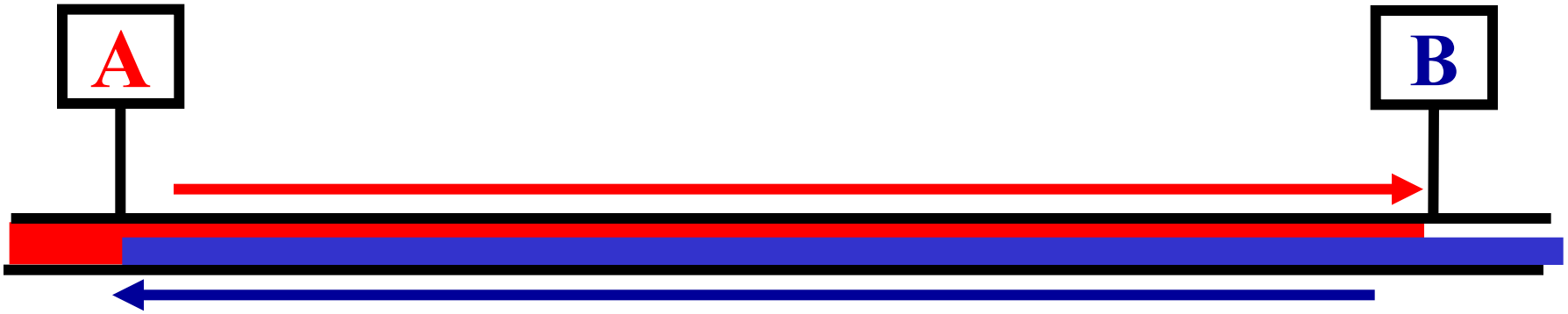






Ethernet Frame Limits

- Upper limit on frames
  - Why?
  - To allow “fair” sharing of the shared bm
- Lower limit on frames
  - Why?
  - To reliably detect collisions
    - For 10Base5, cable max length 2500m → 2 * D_{prop} = 51.2μs
    - transceiver can send & listen at the same time; if received data stream differs from the one transmitted → collision
    - to detect collision: sender must be transmitting when garbled signal propagates back
    - minimum frame = 64bytes = 512bits, take 51μs to transmit at 10Mbps



Ethernet MAC protocol: CSMA/CD

A: sense channel, wait if necessary, when channel is idle, transmit and monitor the channel;

If detect collision

then {

abort and send jam signal;
update collision-count (n++);
delay for K slots (1 slot = 512bits transmission time)
goto A
}

else {finish sending the frame; reset collision-count (n = 0)}

Jam Signal (48 bits): make all other transmitters aware of collision

Exponential Backoff algorithm:

- first collision (n=1): choose K from {0, 1}
- second collision (n =2): choose K from {0, 1, 2, 3}…
- after 10 collisions (n=10), choose K from {0,1,2,3,4,…,1023}

CSMA/CD efficiency

- D_{prop} = max propagation delay between any 2 nodes
- D_{trans} = time to transmit a max-size frame

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

- Efficiency goes to 1
  - as D_{prop} goes to 0
  - as D_{trans} goes to infinity
- much better performance than ALOHA, and still decentralized and simple

Summary of MAC protocols

- Channel partitioning
  - Time Division (TDMA), Frequency Division (FDMA)
- Coordinated access (taking turns)
  - polling from central site or token passing
- Random access
  - ALOHA, S-ALOHA
  - CSMA: Carrier Sensing in MultiAccess: easy in some case (wire), harder in others (wireless)
  - CSMA/CD used in 10Base2, 10Base5 Ethernet
  - CSMA/CA used in 802.11 (WiFi)

Network Links

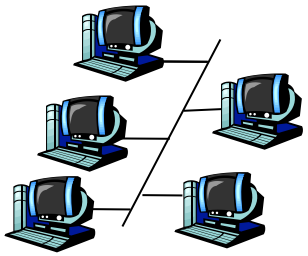
...

- In broadcast links one may want to
  - talk to everybody
  - talk to individual nodes
  - need a some access control

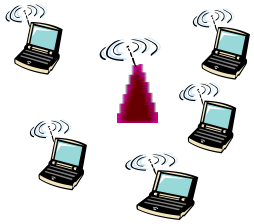
Need addressing

- A set of point-to-point links can appear as broadcast
  - switched Ethernet

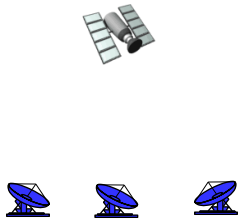
...



shared wire (e.g., cabled Ethernet)



shared RF (e.g., 802.11 WiFi)



shared RF (satellite)

humans at a cocktail party (shared air, acoustical)

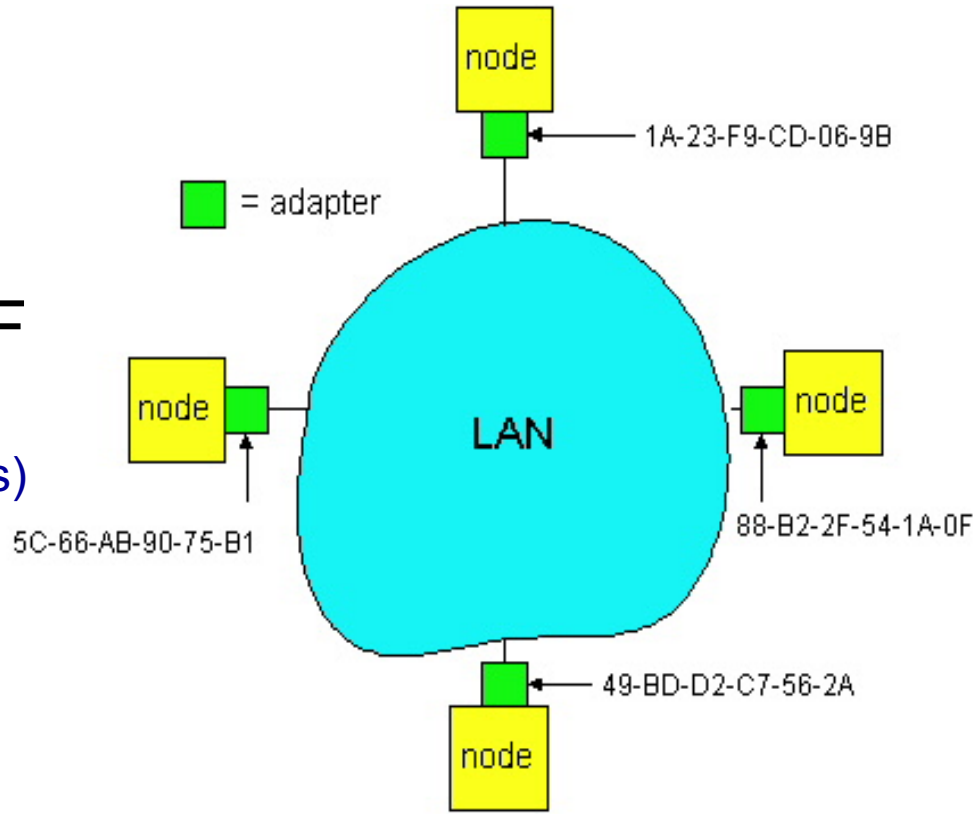
MAC Address (more)

- MAC address allocation by IEEE
- manufacturers buy MAC address blocks from IEEE
- MAC address is flat → portability
  - LAN card can move from one LAN to another
  - In comparison: IP address is hierarchical, NOT portable
    - Tied to the network a node is attached to
- Analogy:
  - MAC address: like Social Security Number
  - IP address: like postal address

MAC (or LAN or Ethernet) Address

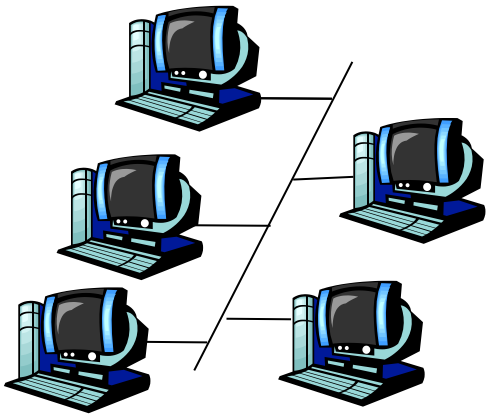
- 48 bit address, used to get data frame from one interface to another physically connected interface (on the same network)
- Each adapter on LAN has a unique LAN address
- Broadcast address:
  FF-FF-FF-FF-FF-FF

  hexadecimal (base 16) notation
  (each "number" represents 4 bits)

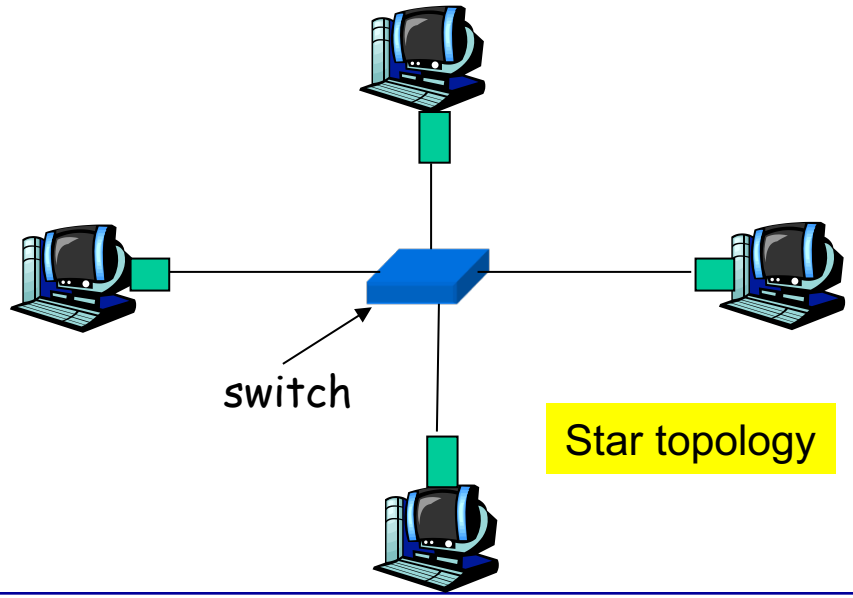


Ethernet

- Dominant LAN technology till wireless showed up
- Kept up with speed race: 10 Mbps – 100 Gbps
- Bus topology popular through mid 90s
- Switch-based star topology prevailed as speed goes up
  - Ethernet protocol runs on each wire

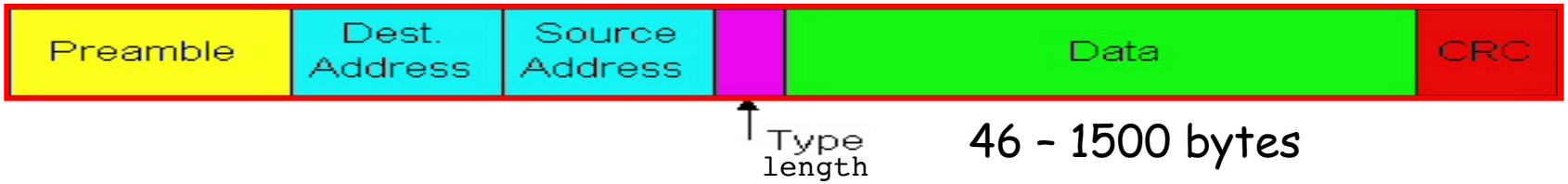


Bus topology



Star topology

Ethernet Frame Structure



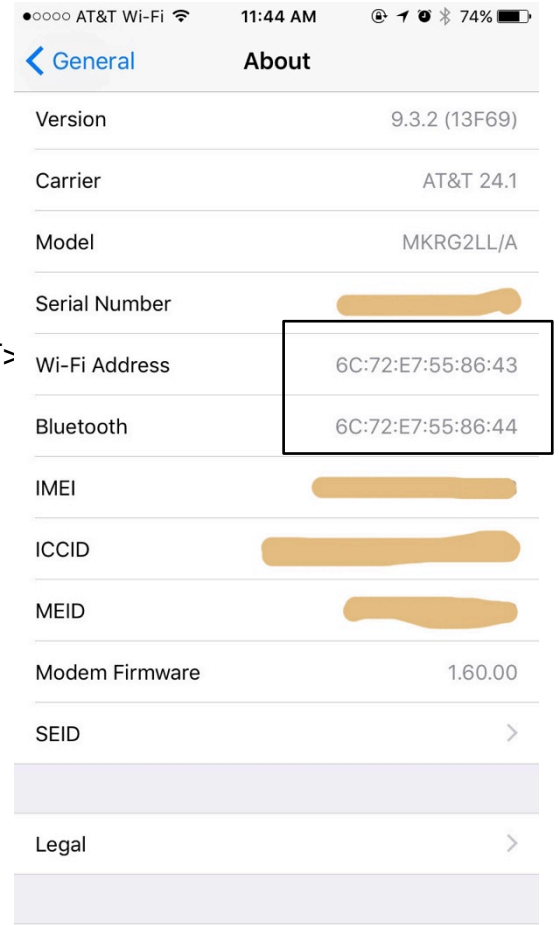
- Sending adapter encapsulates IP datagram in Ethernet frame
- Preamble: 8bytes
  - 7 bytes with pattern 10101010 followed by one byte with pattern 10101011
  - used to synchronize receiver, sender clock rates
- Addresses: 6 bytes each
  - If received frame destination address matches NIC address, or is broadcast address, the adapter passes data to network layer protocol; otherwise, discards frame
- Type: 2 bytes, indicates the higher layer protocol
  - IEEE802.3 changed the “type” field to “length”, defined a separate type field carried in the data part
- CRC: 4 bytes, added by sender, checked at receiver, if error, drop

Ethernet: Unreliable data delivery

- connectionless: No handshaking between sending and receiving NICs
- unreliable: receiving NIC doesn't send acks or nacks to sending NIC
- Ethernet's MAC protocol: CSMA/CD
  - Does not concern with reliable delivery
  - *does concern with collision resolution*

Discovering MAC addresses

- Written on the device (sticker)
- ifconfig (Linux, OS X), ipconfig (Windows)
- Network interface settings




```
✔  11:40 ~ $ ifconfig
en0: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST>
mtu 1500
          ether 3c:15:c2:b8:b1:7e
          inet 169.254.145.207 netmask 0xffff0000 broadcast
169.254.255.255
          nd6 options=1<PERFORMNUD>
          media: autoselect
          status: active
en1: flags=963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX> mtu 1500
          options=60<TSO4,TSO6>
          ether 72:00:03:04:73:00
          media: autoselect <full-duplex>
          status: inactive
en2: flags=963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX> mtu 1500
          options=60<TSO4,TSO6>
          ether 72:00:03:04:73:01
          media: autoselect <full-duplex>
          status: inactive
```

Discovery Info About MAC Address

OUI search

3c:15:c2:b8:b1:7e
72:00:03:04:73:00
72:00:03:04:73:01
68:5b:35:c0:61:b6
6c:72:e7:55:86:44

Find

Results

3C:15:C2 Apple, Inc.
68:5B:35 Apple, Inc.
6C:72:E7 Apple, Inc.

https://www.wireshark.org/tools/oui-lookup.html

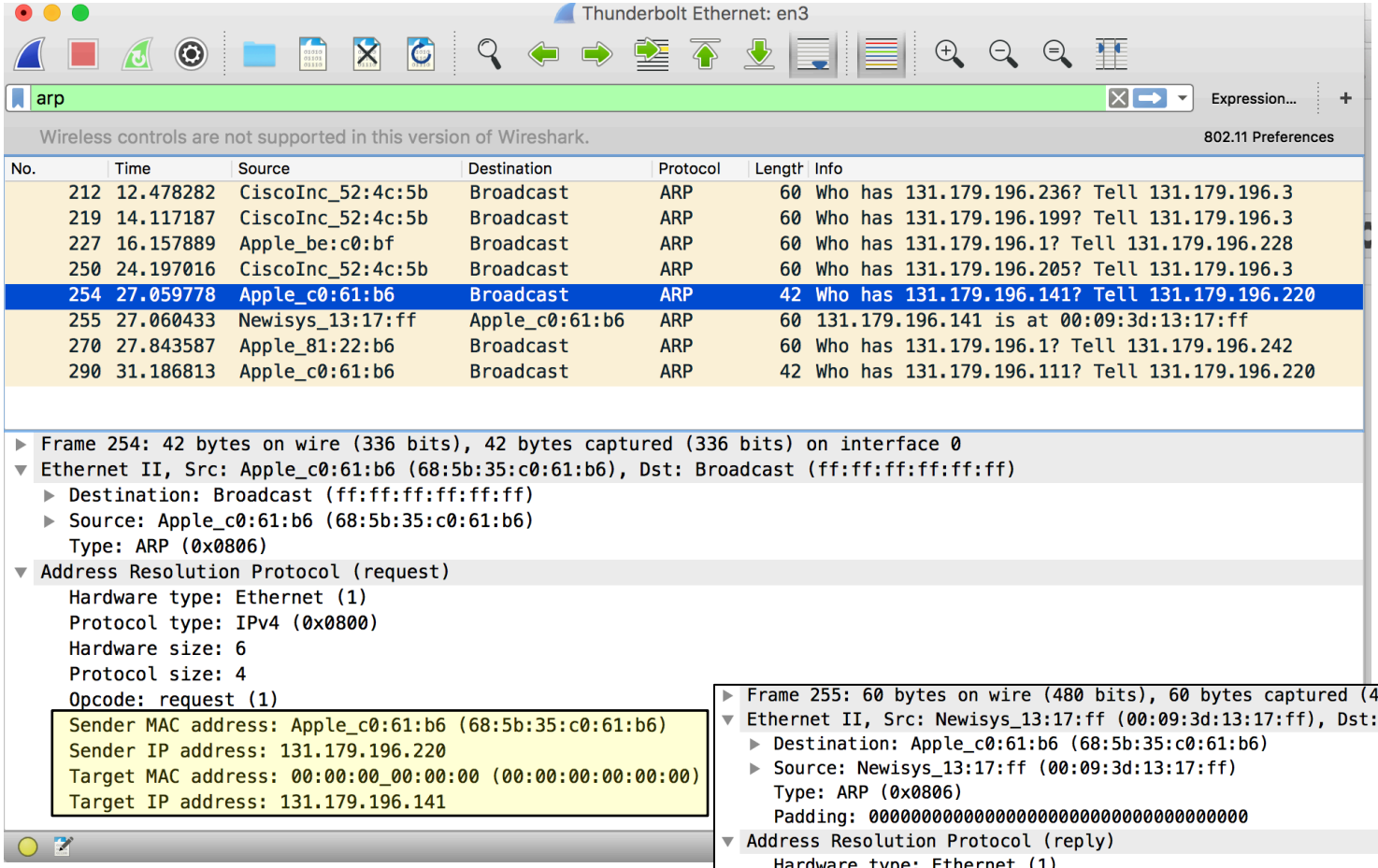
Packet Delivery Dilemma

- Link layer uses MAC addresses to deliver packets to destinations
  - unique within LAN segment
- IP router knows only IP address of the destination?
  - globally unique*
- How to deliver packets?

Address Resolution Protocol (ARP)

- A knows B's IP address, wants to learn B's MAC address
- A *broadcasts* ARP query msg containing B's IP address
  - Dest. MAC address: FF-FF-FF-FF-FF-FF (broadcast)
  - all machines on LAN receive ARP query
- B receives ARP query, replies with B's MAC address
  - sent to A's MAC address (unicast)
- A caches B's [IP:MAC] address pairs until the information becomes too old

ARP Example with Wireshark



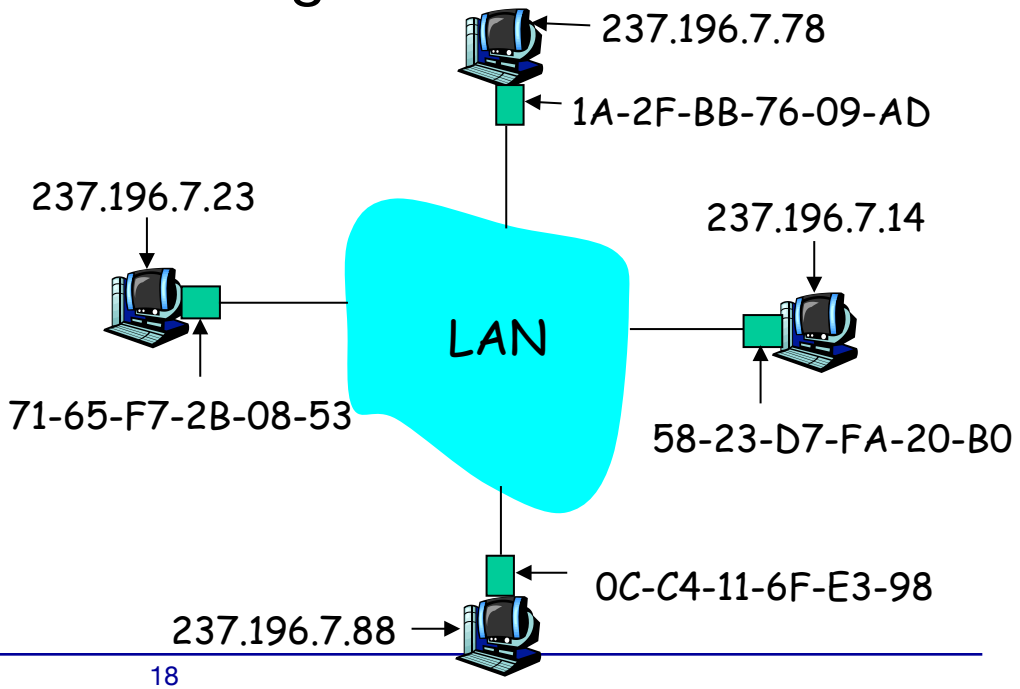
ARP: plug-and-play

- Every IP node (Host, Router) on LAN keeps ARP table
- Each table entry: represents a node on the LAN:

  < IP address; MAC address; TTL>

  - TTL (Time To Live): the entry's lifetime (typically 20 min)
  - One example of *soft-state* design: information deletes itself after certain time unless being refreshed

Nodes create their ARP tables without intervention from network administrator



Look Into Your ARP Table

✔ **12:06 ~ OSX $ arp -an**

? (131.179.196.1) at 0:0:c:7:ac:c4 on en3 ifscope [ethernet]

? (131.179.196.69) at 0:8:74:92:b3:8a on en3 ifscope [ethernet]

? (131.179.196.255) at (incomplete) on en3 ifscope [ethernet]

? (169.254.255.255) at (incomplete) on en3 [ethernet]

? (192.168.2.255) at (incomplete) on bridge100 ifscope [bridge]

? (224.0.0.251) at 1:0:5e:0:0:fb on en3 ifscope permanent [ethernet]

? (255.255.255.255) at (incomplete) on en3 ifscope [ethernet]

[cawka@benz LINUX ~]$ arp -an

? (131.179.196.47) at 00:09:3d:13:17:e8 [ether] on br0

? (131.179.196.10) at 30:46:9a:11:7a:14 [ether] on br0

? (131.179.196.69) at 00:08:74:92:b3:8a [ether] on br0

? (131.179.196.46) at 00:1d:09:94:65:38 [ether] on br0

Administrators can also modify ARP entries, though not normally needed

Sending packets: $IP^A \rightarrow IP^B$

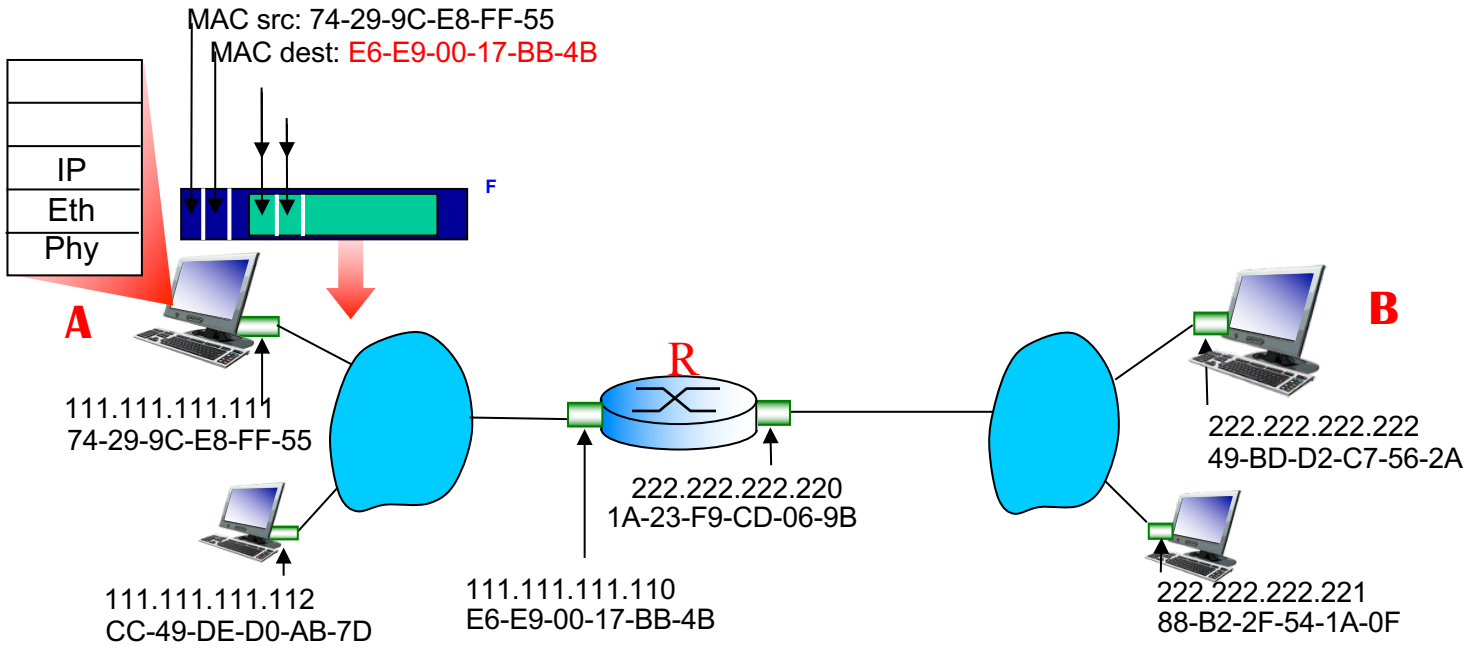
- Find an entry in the routing table

```
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
131.179.196.0   0.0.0.0         255.255.255.0   U     0      0        0 br0
```

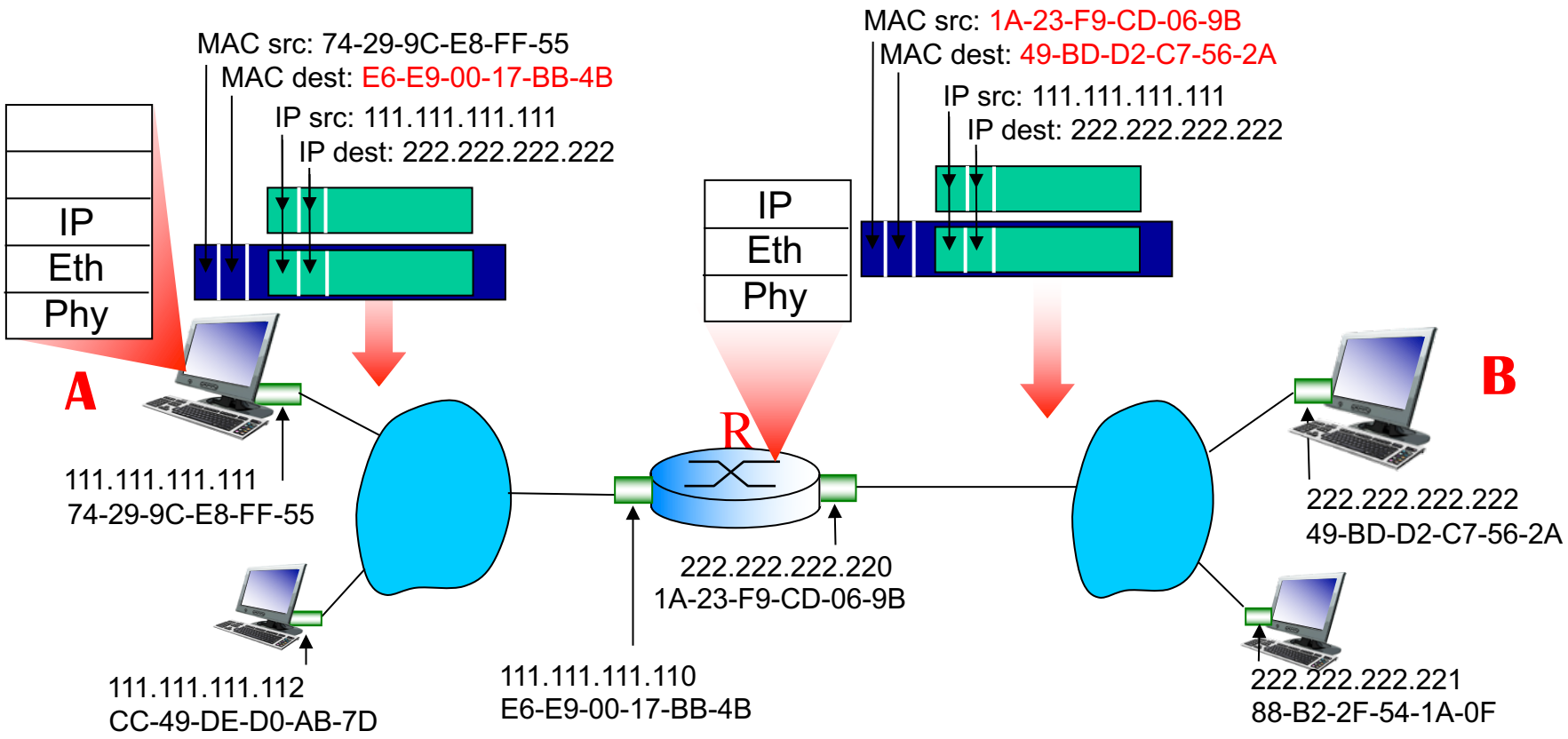
- If entry is saying that packet can be sent directly
  - Lookup MAC for destination IP (can be cached)

```
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         131.179.196.1   0.0.0.0         UG    0      0        0 br0
```

- If entry is saying that packet should be set to the gateway
  - Lookup MAC for the gateway's IP
- Create frame with the found MAC and original IP packet as a payload
- Send the frame



- Router or node receives the frame, as it is destined to it
- Removes Ethernet header, finds IP destination address
- If IP is “self”, deliver to transport, which will deliver to the app
- If IP is not self and node is router, repeat the previous steps (lookup routing table, lookup ARP, ...)

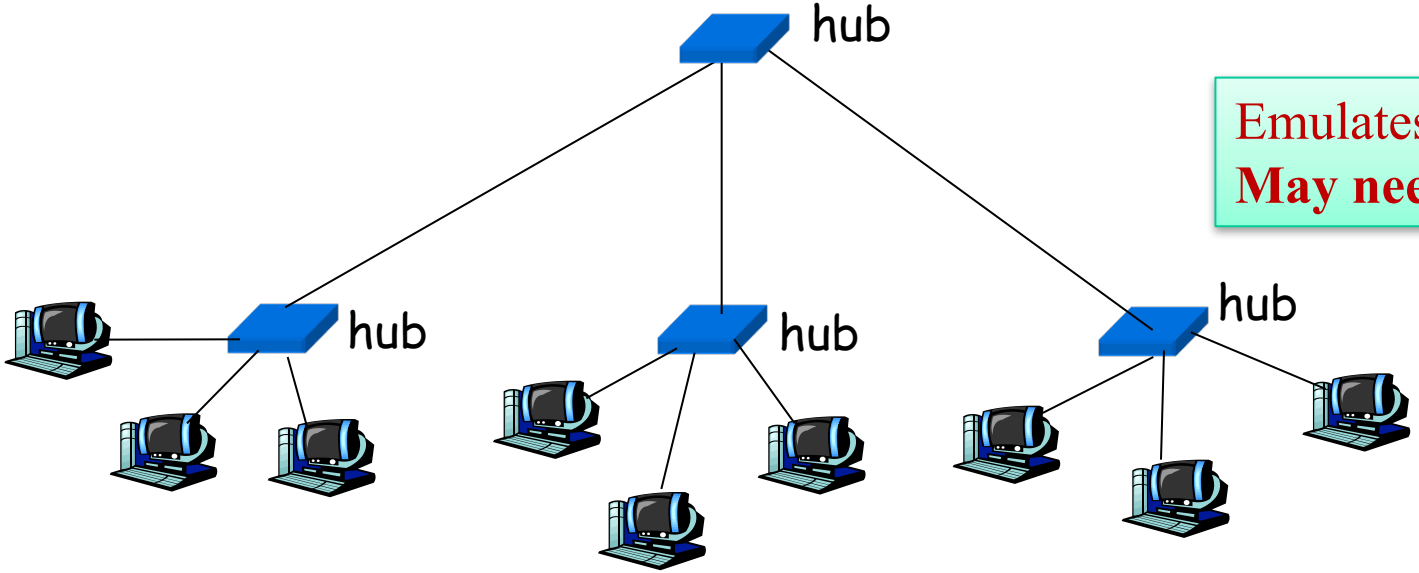


Ethernet Interconnection

Repeaters, Hubs, Switches

Ethernet Hub and Repeaters

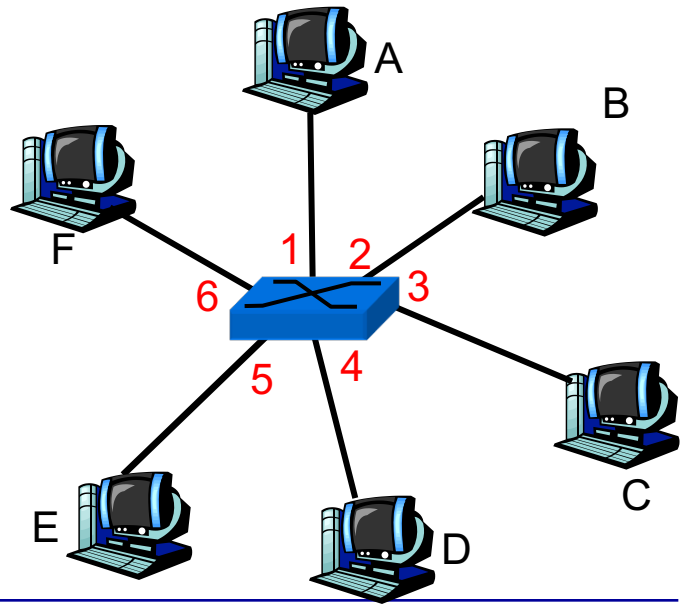
- Repeaters: physical layer devices
  - bits coming in one link go out all other links at same rate
- Hubs: link layer devices
  - packets coming in one port/link, go out all other ports/links
- Both
  - Extend max distance between nodes
  - Transmission by one node may collide with any node residing at any segment connected to the same hub
  - All pieces connected by one hub considered as one Ethernet



Emulates bus topology
May need CSMA/CD

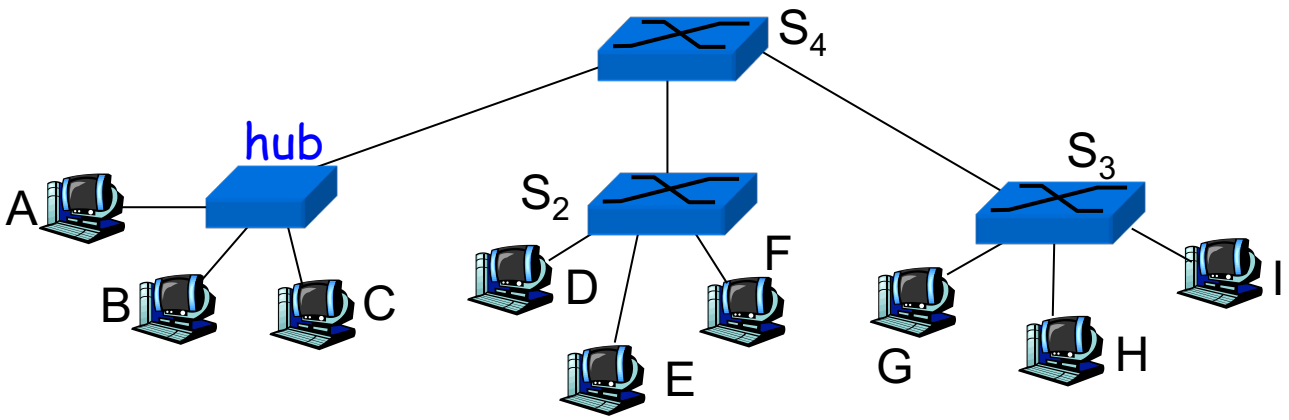
Ethernet Switch

- ***Link layer device***: behave as a host on each connected LAN
  - speak Ethernet protocol at each interface
  - knows ARP protocol, but just listens and acts (no actively speaking)
  - transparent: hosts are unaware of presence of switches
- Store-and-forward: examine each incoming frame's MAC address, forward to the destination LAN if dest. host is on a different LAN
  - A→D and B→E can go simultaneously
  - buffer frames temporarily if next LAN busy
- *NO configuration needed*
  - Plug-and-play



Switches vs. hubs

- Bit forwarding vs store-and-forward devices
  - hub: pure signal amplification
    - e.g. if A sends a frame, S4 can hear it
  - switches: buffer then forward
    - e.g. if D sends a frame to E, S4 does not hear it
- Switch can connect different LANs



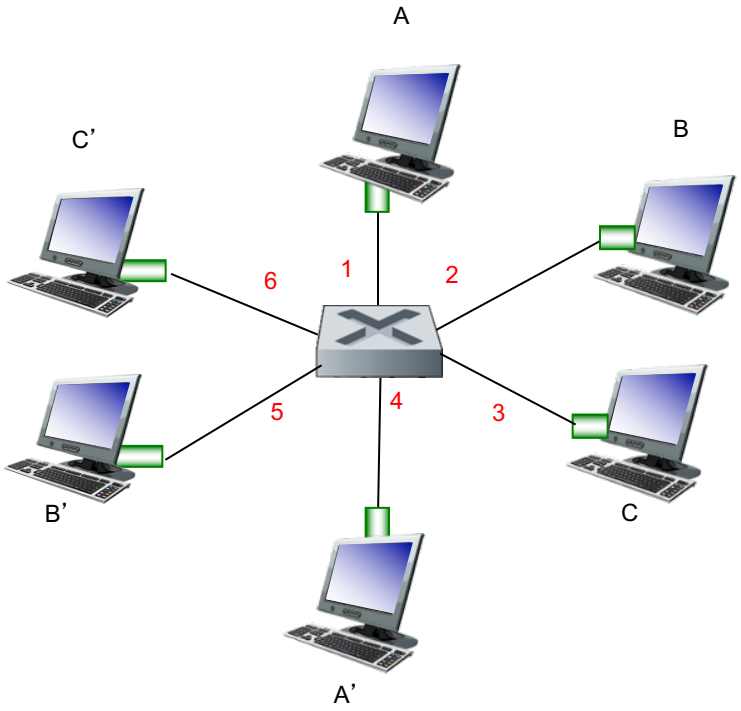
Switch needs a forwarding table

Q: how does switch know A is reachable via interface 4, B is reachable via interface 5?

- A: each switch has a switch table, each entry:
  - (MAC address of host, interface to reach host, time stamp)
  - looks like a routing table!

Q: how are entries created, maintained in switch table?

- something like a routing protocol?



An ethernet switch with 6 Interfaces (1,2,3,4,5,6)

Building a forwarding table by self-learning

When receive data frame:

1. If not already in the table: record incoming link, MAC address of sending host
   - Each entry contains: MAC address, Interface, time stamp
   - stale entries are dropped (**TTL can be 60 min**)
2. switch table indexed by MAC destination address
3. if table entry found for destination
   then {

   if destination on segment from which frame arrived
   then drop frame

   else forward frame to interface indicated by entry

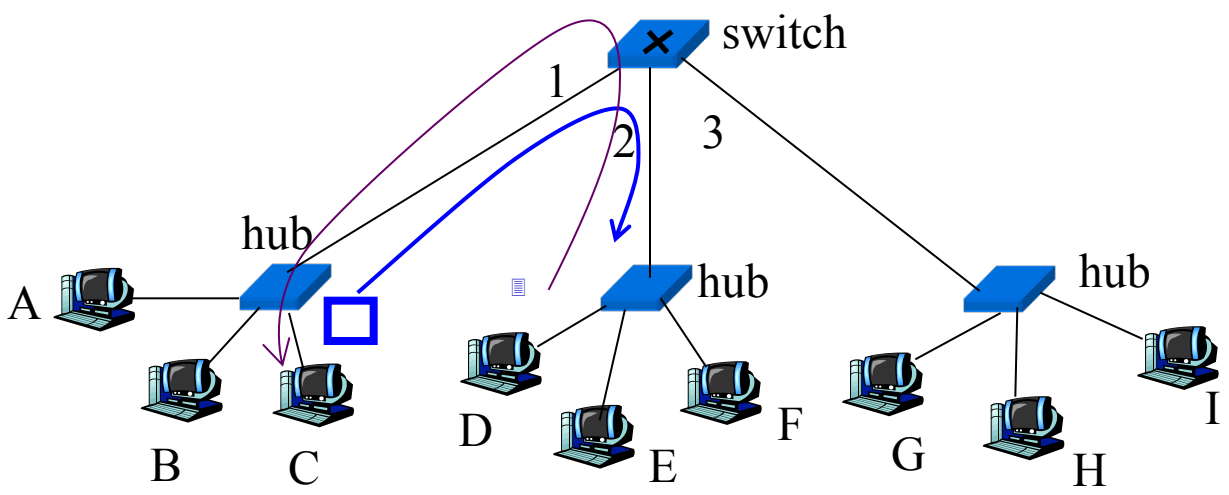
   }

   else flood /* forward to all interfaces except the arriving interface */

Ethernet switch example

important

Suppose C sends a data frame ☐ to D



address	interface
A	1
B	1
E	2
G	3
C	1
D	2

Switch receives frame ☐ from C

- Add to forwarding table: C is on interface 1
- D is not in table: forwards ☐ to interfaces 2 and 3

☐ is received by D

When D replies back with a frame 🗎 to C

Switch: Add to forwarding table: D is on interface 2

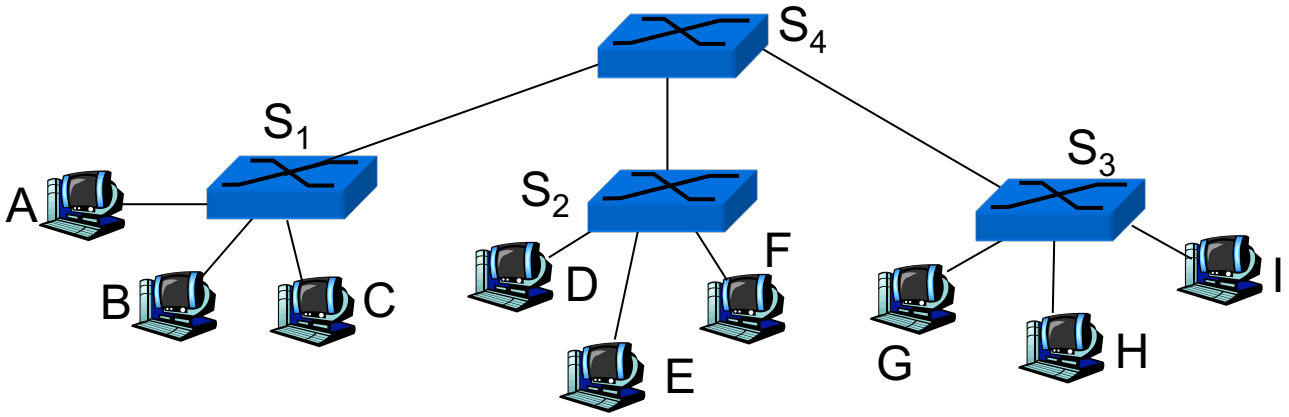
Forward 🗎 to C

Interconnecting switches

- Switches can be connected together

Q: sending from A to F -- how does S_1 know to forward frame destined to F via S_4 and S_2?

A: self learning! (works exactly the same as in single-switch case)



Self-learning switches: an example

- Consider the following frame transmission:

  A → B, B → A

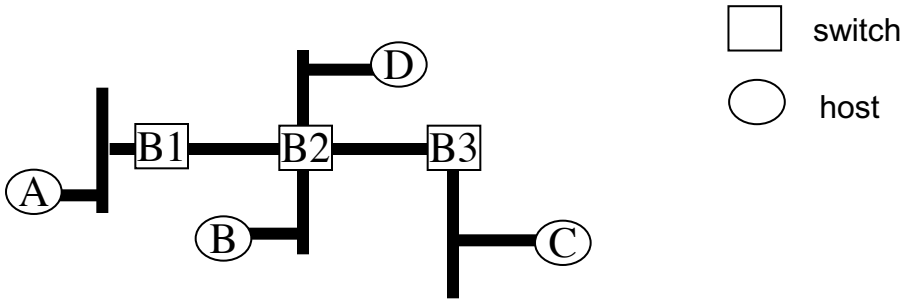
  B → C, C → B

  D → C, C → D

Q: How many times does B2 need to flood the data frame?

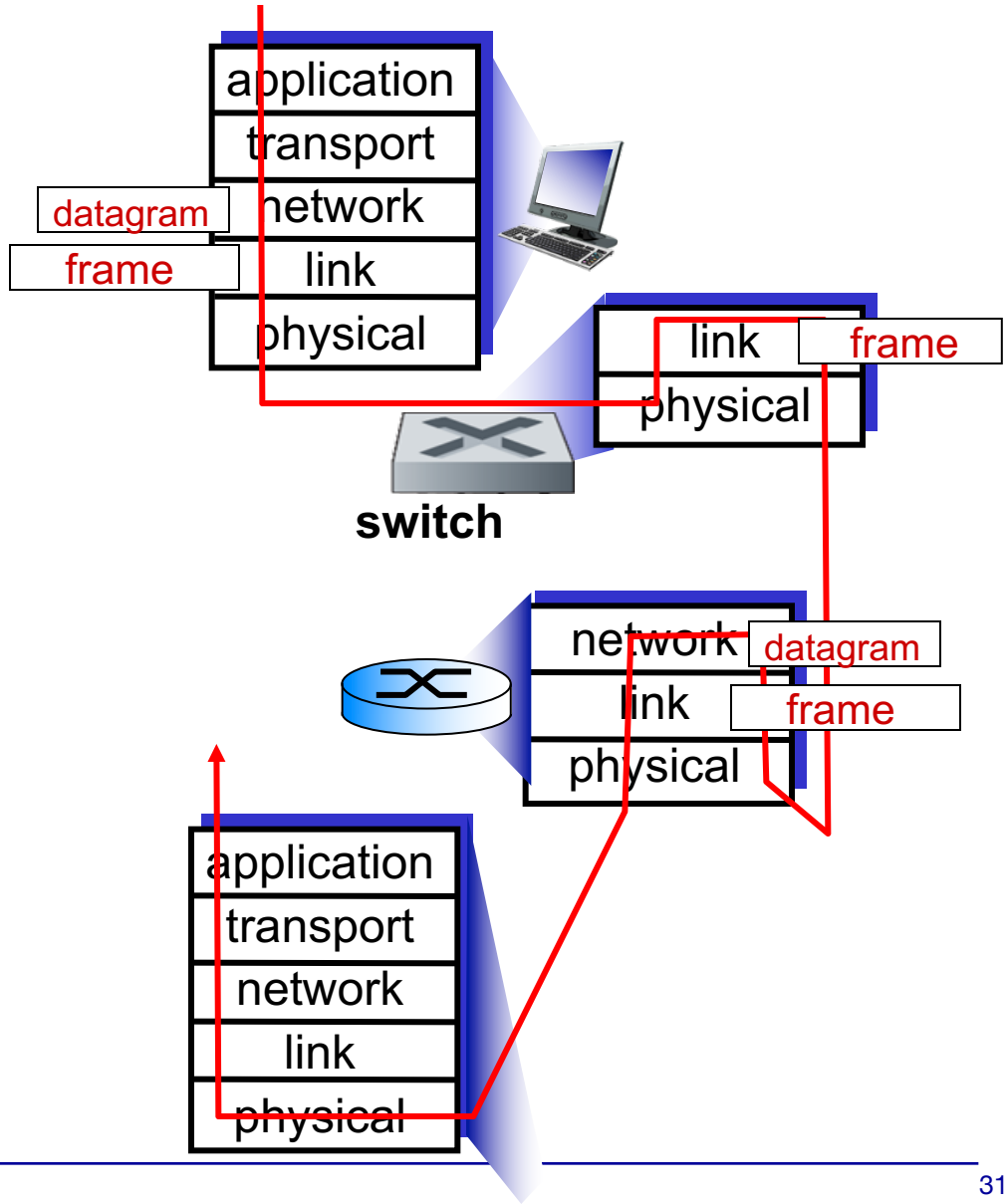
A→B: B2 flood

B→C: B2 flood



Switches vs. Routers

- both are store-and-forward devices
  - routers: network layer devices (examine IP headers)
  - Switches: link layer devices (examine Ethernet headers)
- Build forwarding table
  - routers run routing protocols
  - switches implement self-learning algorithms



Switches: advantages and limitations

- Transparent: no change to hosts
- Isolates collision domains
  - resulting in higher total maximum throughput
- Can connect Ethernets of different speeds
  - because it is a store and forward device
- Constrained topology: ***tree only***
  - all inter-segment traffic concentrated on a single tree
  - (all multicast traffic forwarded to all LAN's)

Routers: advantages and limitations

- Support arbitrary topologies
- Efficient support for multicast routing
- Require IP address configuration (not plug-and-play)
- More complex packet processing than switches
- Switches do well in small setting, routers are used in large networks

Q: what is in a routing table, an ARP table, and a switch's forwarding table?

Interconnecting LANs

Q: Why not just one big LAN?

- Signal attenuation; limited physical distance
- all stations on one LAN share bandwidth → limited total throughput

- A number of different types of "connectors"
  - router (already mentioned, L3)
  - bridge, Ethernet switch (L2)
    - Frame forwarding
    - Perform CSMA/CD
  - repeater, hub (L1)
    - bit level forwarding
    - Signal amplification