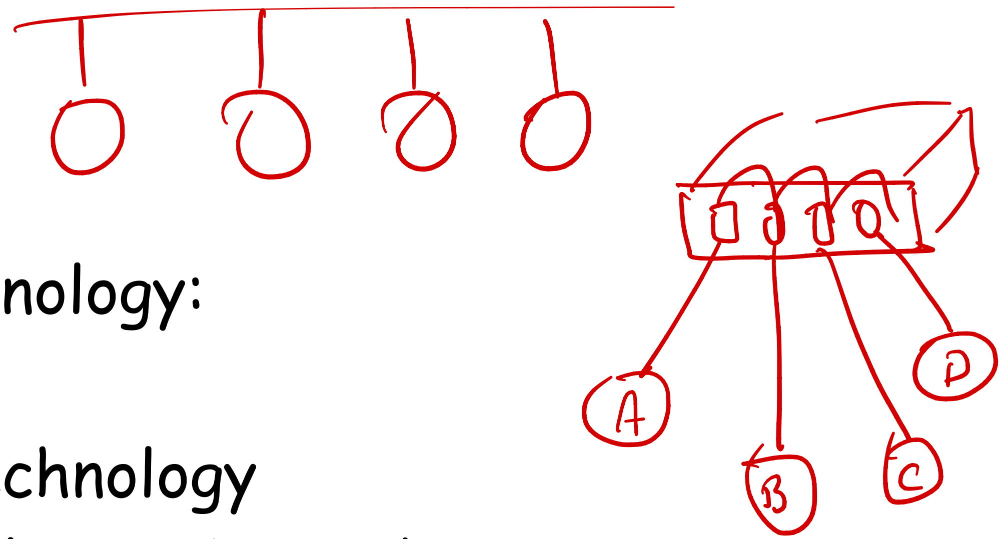
# Link Layer

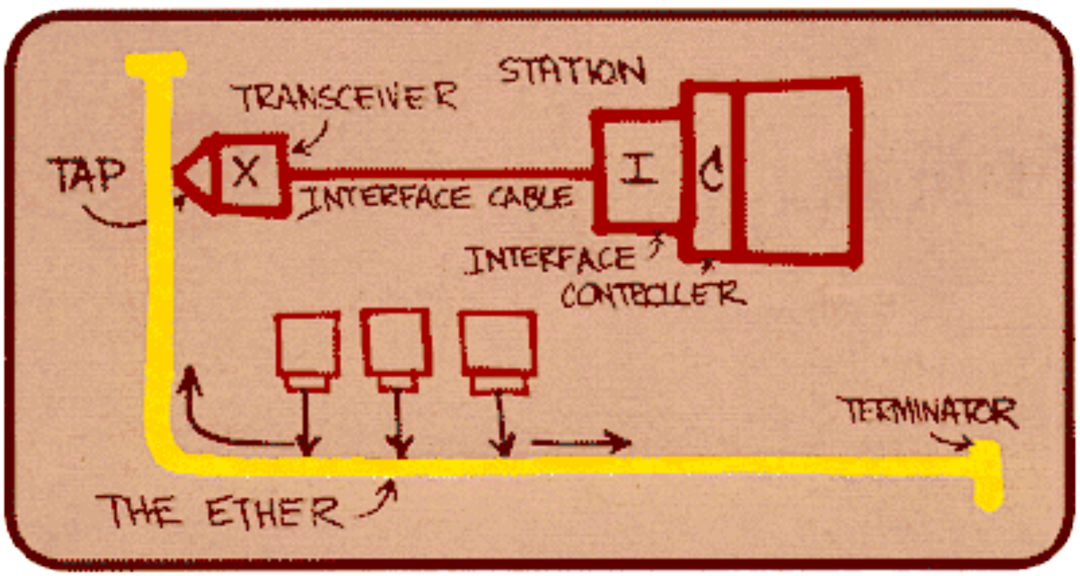
- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3Multiple access protocols
- 5.4 Link-Layer Addressing
- 5.5 Ethernet
- 5.6 Hubs and switches
- 5.7 PPP
- 5.8 Link Virtualization: ATM

# Ethernet




"dominant" wired LAN technology:

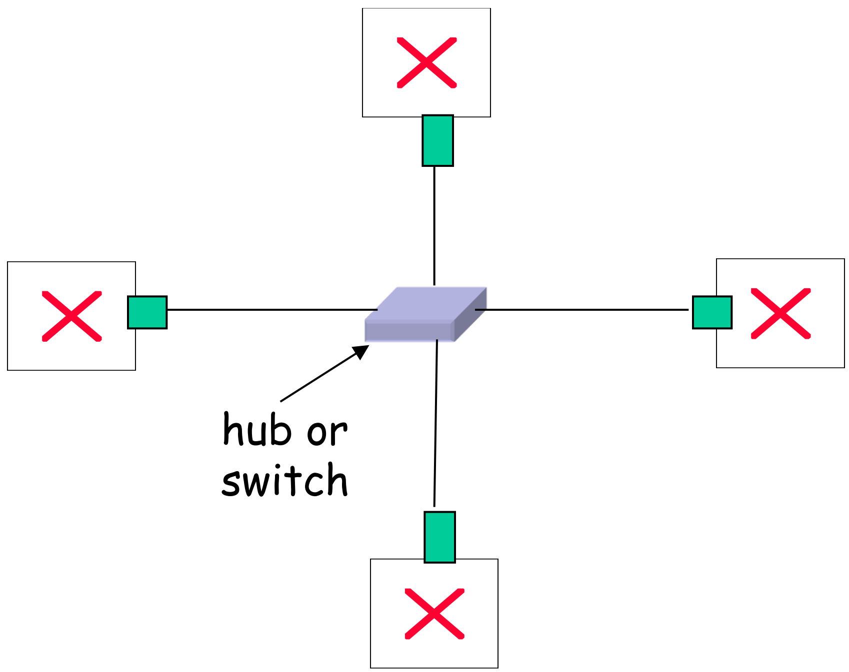
- cheap $20 for 100Mbs!
- first widely used LAN technology
- Simpler, cheaper than token LANs and ATM
- Kept up with speed race: 10 Mbps – 10 Gbps




Metcalfe's Ethernet sketch

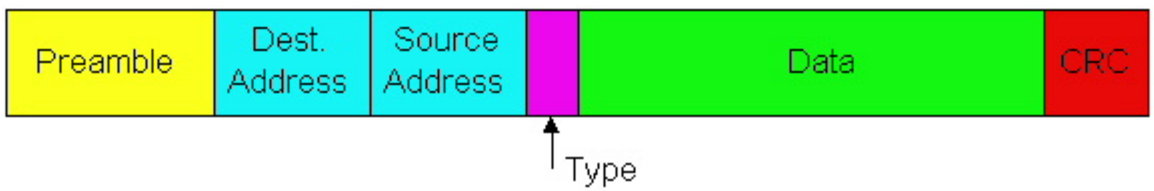
# Star topology

- Bus topology popular through mid 90s
- Now star topology prevails
- Connection choices: hub or switch (more later)



# Ethernet Frame Structure

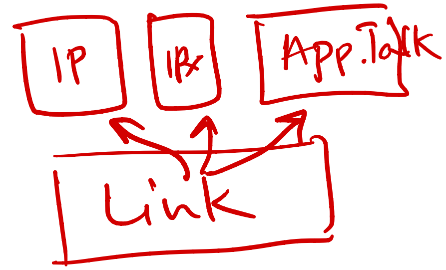
Sending adapter encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame



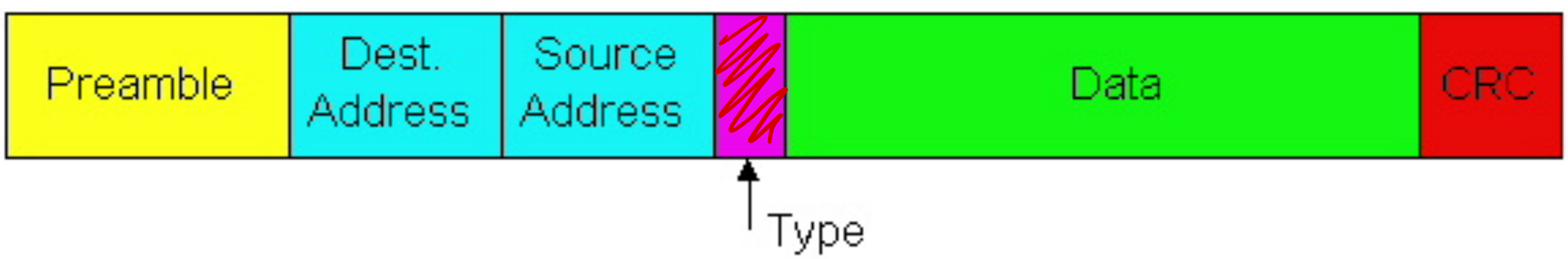

Preamble:

- 7 bytes with pattern 10101010 followed by one byte with pattern 10101011
- used to synchronize receiver, sender clock rates

# Ethernet Frame Structure (more)




- **Addresses:** 6 bytes
  - if adapter receives frame with matching destination address, or with broadcast address (eg ARP packet), it passes data in frame to net-layer protocol
  - otherwise, adapter discards frame
- **Type:** indicates the higher layer protocol (mostly IP but others may be supported such as Novell IPX and AppleTalk)
- **CRC:** checked at receiver, if error is detected, the frame is simply dropped



# Unreliable, connectionless service

- Connectionless: No handshaking between sending and receiving adapter.
- Unreliable: receiving adapter doesn't send acks or nacks to sending adapter
  - stream of datagrams passed to network layer can have gaps
  - gaps will be filled if app is using TCP
  - otherwise, app will see the gaps

# Ethernet uses CSMA/CD

Collision Detection

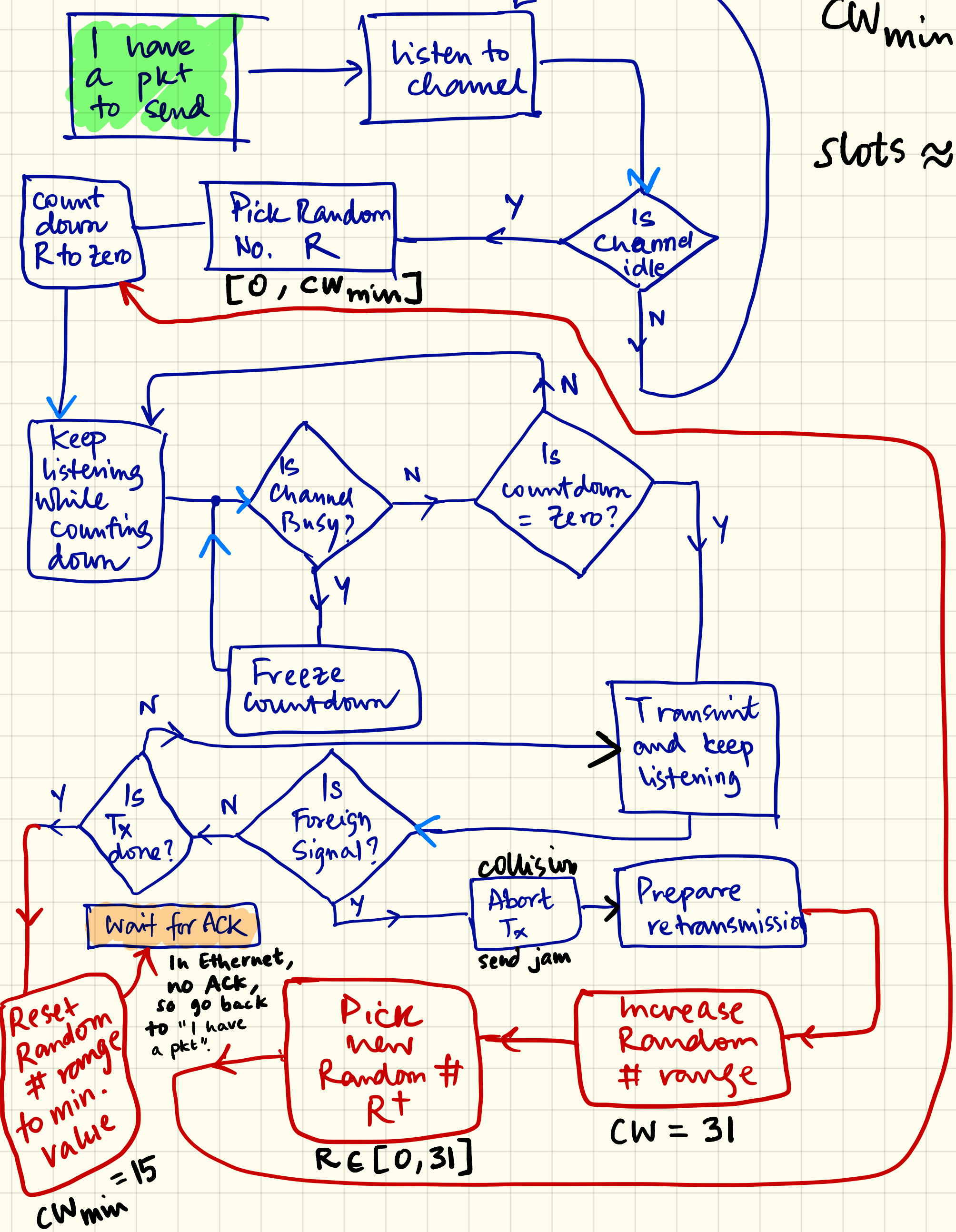
- No slots
- adapter doesn't transmit if it senses that some other adapter is transmitting, that is, carrier sense
- transmitting adapter aborts when it senses that another adapter is transmitting, that is, collision detection
- Before attempting a retransmission, adapter waits a random time, that is, random access

So current protocol:

Contention Window:

$CW_{min} = 15$ slots

slots $\approx 10\,\mu s$.



(→) Some key points.

① Collision happens always at the Receiver. Transmitter may detect collision by observing a foreign signal, but that doesn't mean collision is at $T_x$.

② Channel is wasted because of random count down ⇒ called BACKOFF. This is the price to be paid for distributed coordination.

③ The above protocol assumes that a $T_x$ can transmit and listen at the same time. Possible in wired networks like Ethernet. Harder in wireless networks.

④ $T_x$ detects foreign signal and can tell for sure that collision is happening at $R_x$. This assumes channel is identical at $T_x$ and $R_x$. True for wired networks, not for wireless.

# Ethernet CSMA/CD algorithm

1. Adaptor receives datagram from net layer & creates frame
2. If adapter senses channel idle, it starts to transmit frame. If it senses channel busy, waits until channel idle and then transmits
3. If adapter transmits entire frame without detecting another transmission, the adapter is done with frame !
4. If adapter detects another transmission while transmitting, aborts and sends jam signal
5. After aborting, adapter enters **exponential backoff**: after the mth collision, adapter chooses a K at random from $\{0,1,2,\ldots,2^m-1\}$. Adapter waits K·512 bit times and returns to Step 2

Can we use the same concepts from Ethernet in wireless?

such as WiFi ?

# Link Layer

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3Multiple access protocols
- 5.4 Link-Layer Addressing
- 5.5 Ethernet
- 5.6 Hubs and switches

# MAC Addresses and ARP

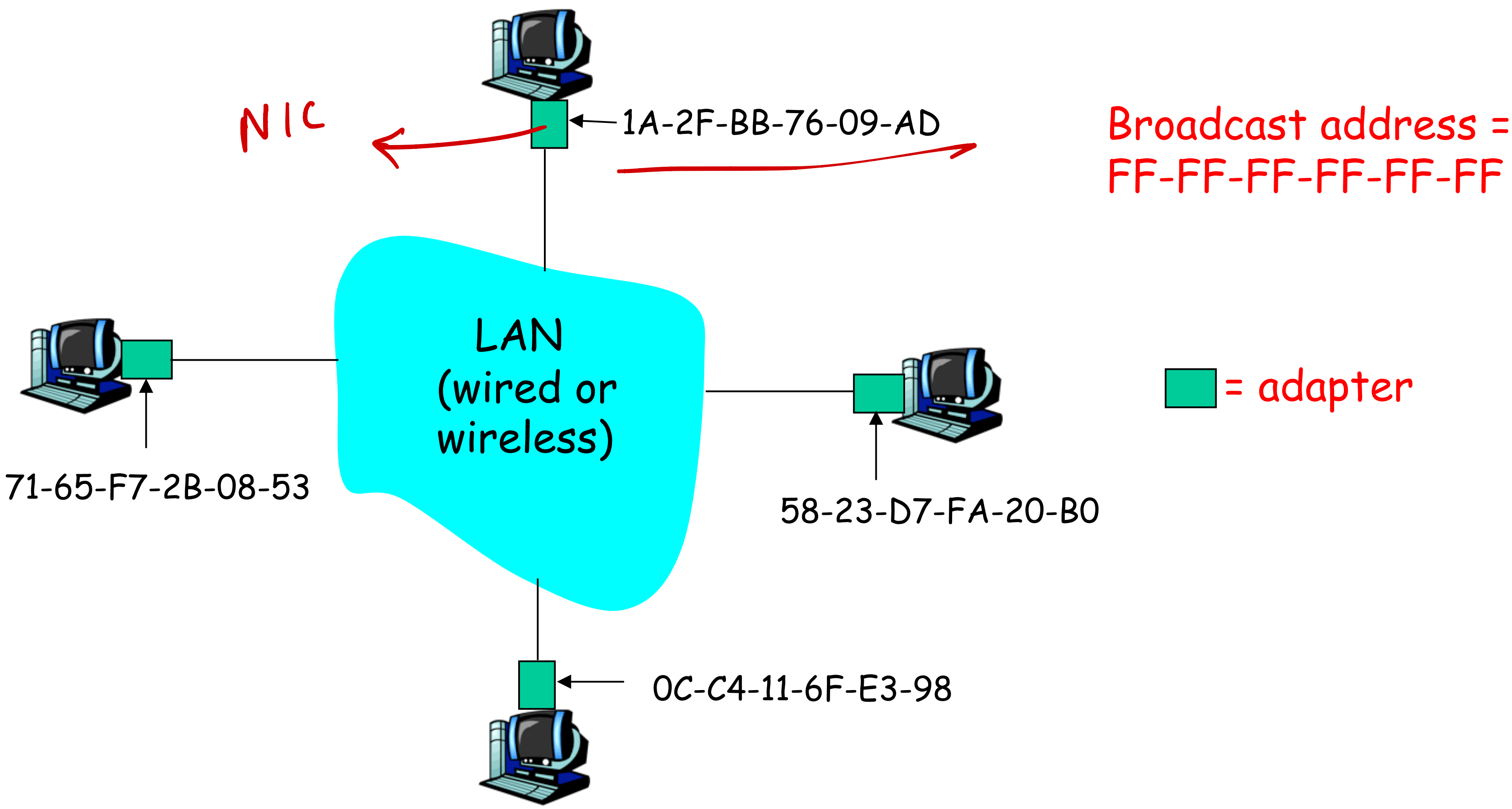
Network Interface Card (NIC)

Eth card

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

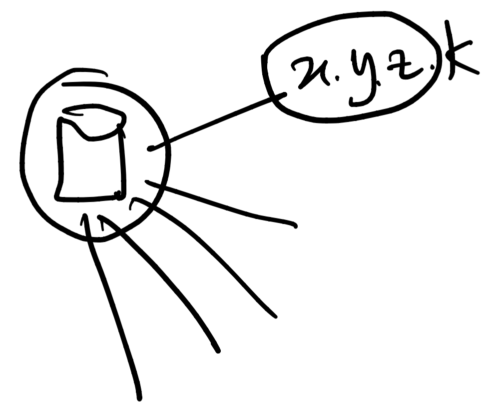
# LAN Addresses and ARP

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:
  - (a) MAC address: like Social Security Number
  - (b) IP address: like postal address
- MAC flat address ➜ portability
  - can move LAN card from one LAN to another
- IP hierarchical address NOT portable
  - depends on IP subnet to which node is attached

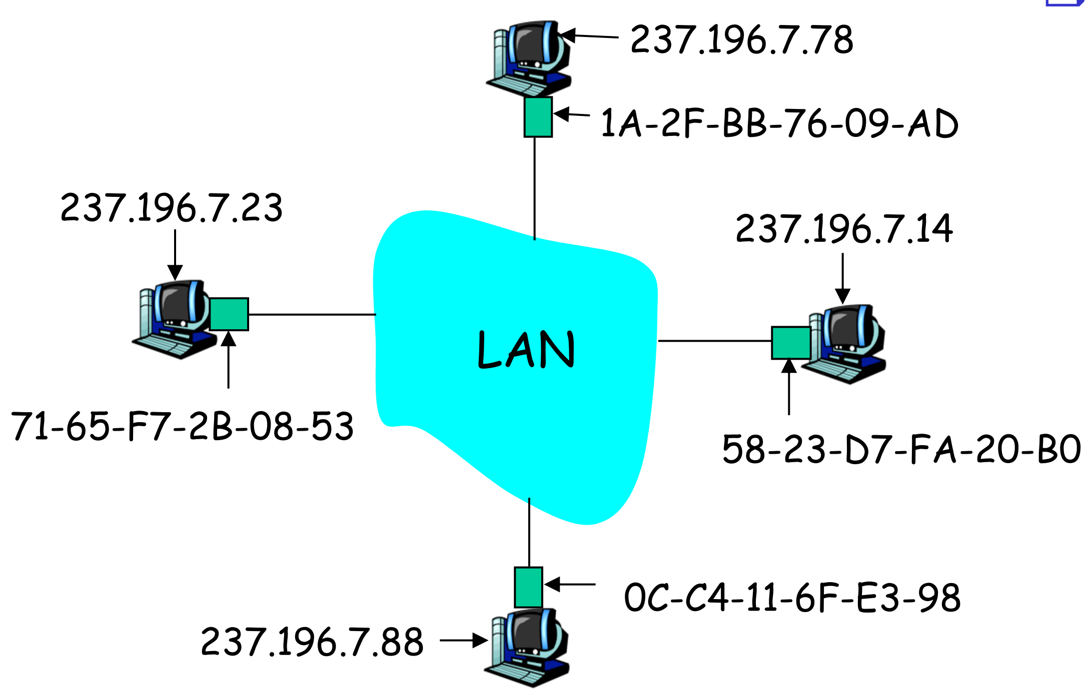



x.y.z.k

x.y.z.k/28

# ARP: Address Resolution Protocol

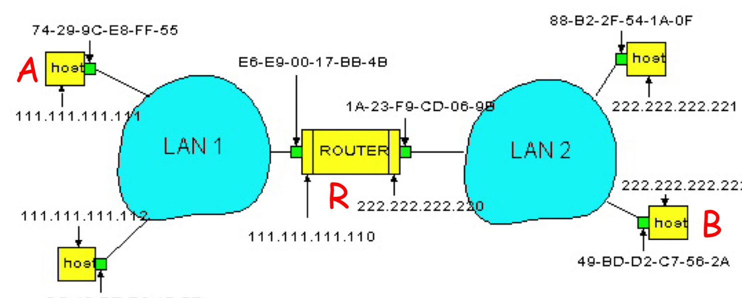
Question: how to determine MAC address of B knowing B's IP address?




- 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)

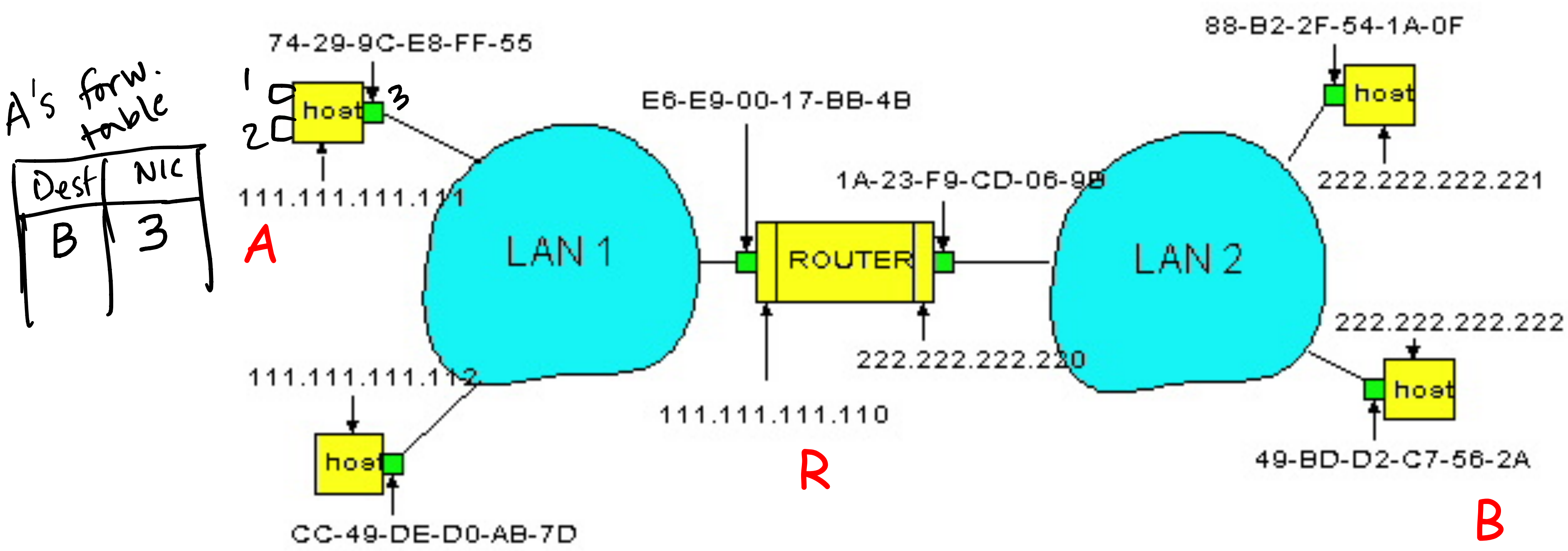
# ARP protocol: Same LAN (network)

- 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



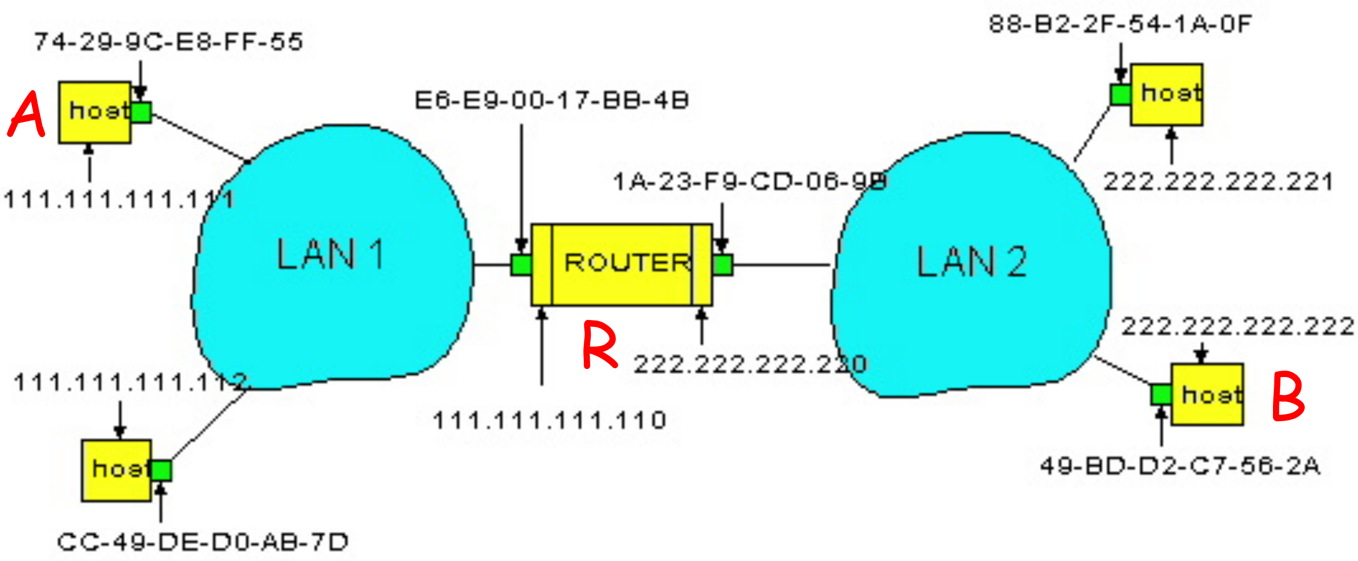
# Routing to another LAN

walkthrough: send datagram from A to B via R
assume A know's B IP address




- Two ARP tables in router R, one for each IP network (LAN)

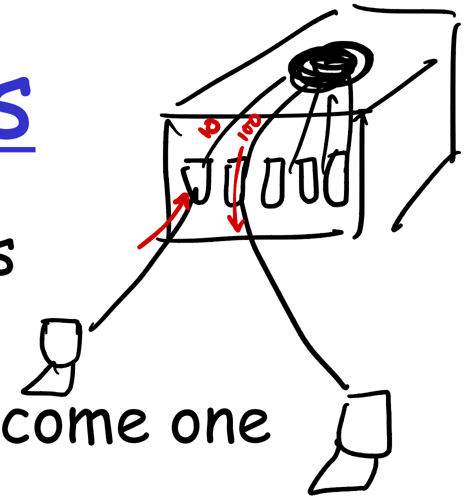
- A creates datagram with source A, destination B
- A uses ARP to get R's MAC address for 111.111.111.110
- A creates link-layer frame with R's MAC address as dest, frame contains A-to-B IP datagram
- A's adapter sends frame
- R's adapter receives frame
- R removes IP datagram from Ethernet frame, sees its destined to B
- R uses ARP to get B's MAC address
- R creates frame containing A-to-B IP datagram sends to B



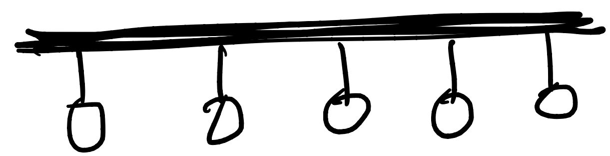
# Link Layer

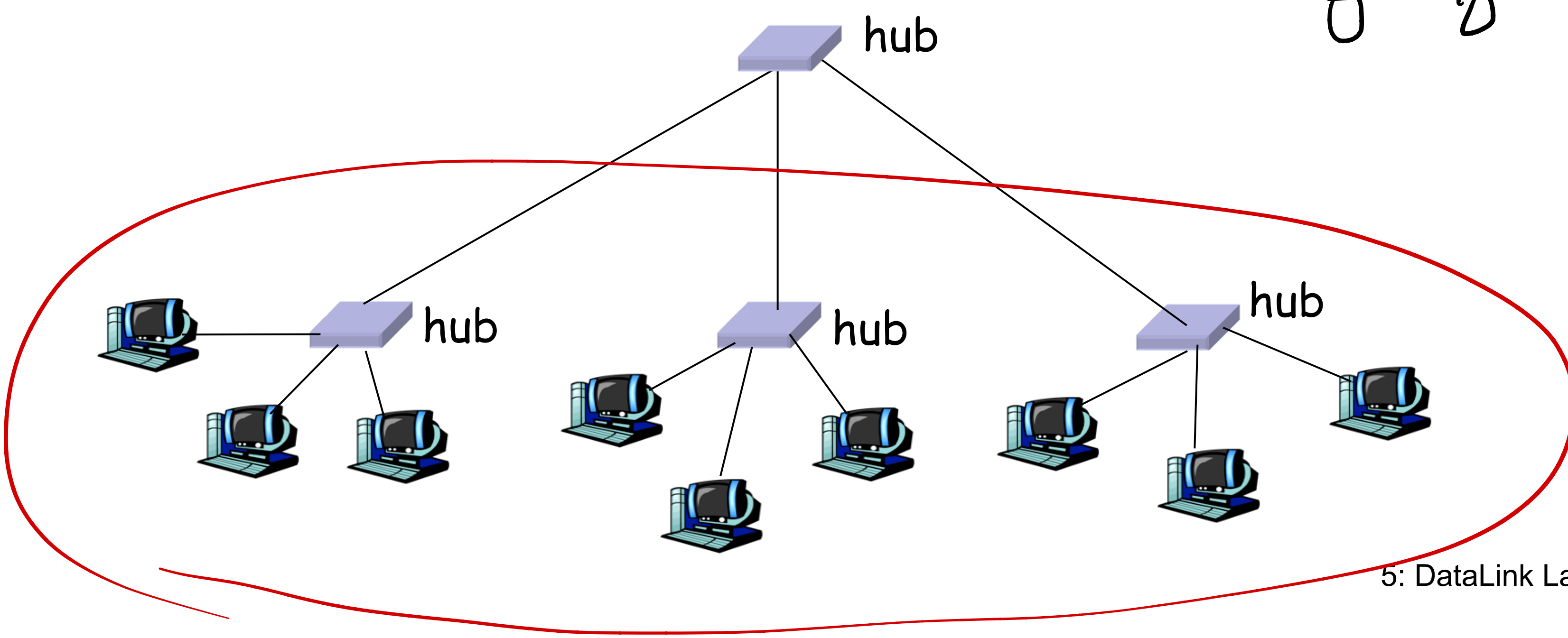
- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3Multiple access protocols
- 5.4 Link-Layer Addressing
- 5.5 Ethernet
- 5.6 Interconnections: Hubs and switches

# Interconnecting with hubs



- Backbone hub interconnects LAN segments
- Extends max distance between nodes
- But individual segment collision domains become one large collision domain
- Also, can't interconnect 10BaseT & 100BaseT

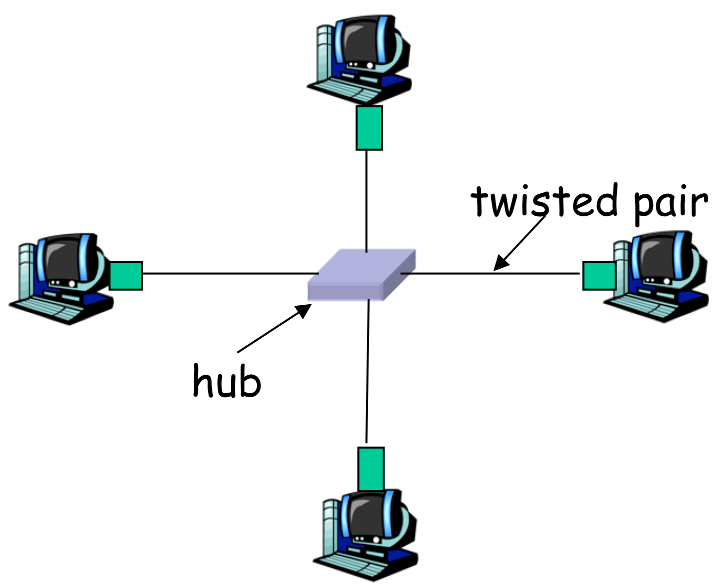




# Hubs

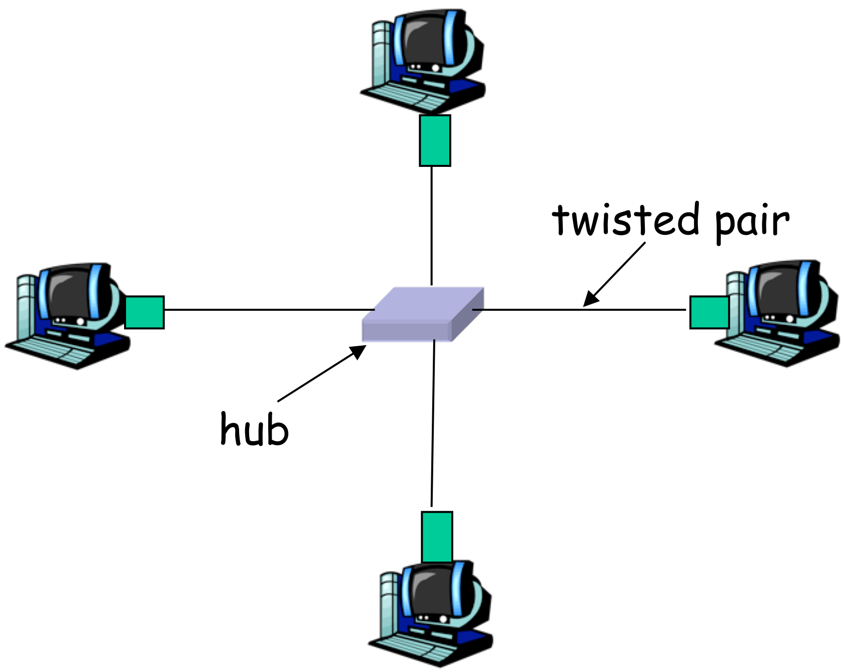
Hubs are essentially physical-layer repeaters:

- bits coming from one link go out all other links
- at the same rate
- no frame buffering
- no CSMA/CD at hub: adapters detect collisions
- provides net management functionality



# 10BaseT and 100BaseT

- 10/100 Mbps rate; latter called "fast ethernet"
- T stands for Twisted Pair
- Nodes connect to a hub: "star topology"; 100 m max distance between nodes and hub



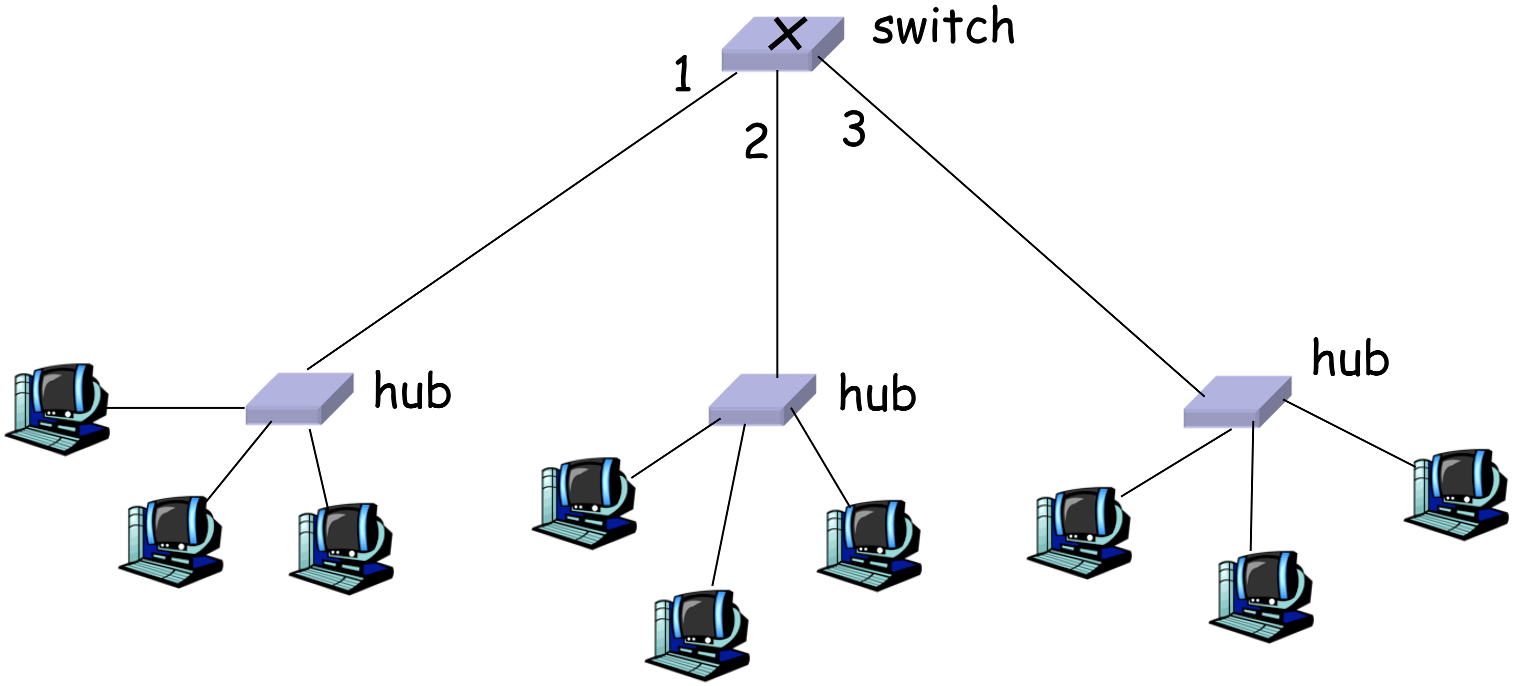
# Gbit Ethernet

- uses standard Ethernet frame format
- allows for point-to-point links and shared broadcast channels
- in shared mode, CSMA/CD is used; short distances between nodes required for efficiency
- uses hubs, called here "Buffered Distributors"
- Full-Duplex at 1 Gbps for point-to-point links
- 10 Gbps now !

# Switch

- Link layer device
  - stores and forwards Ethernet frames
  - examines frame header and selectively forwards frame based on MAC dest address
  - when frame is to be forwarded on segment, uses CSMA/CD to access segment
- transparent
  - hosts are unaware of presence of switches
- plug-and-play, self-learning
  - switches do not need to be configured

# Forwarding




- How do determine onto which LAN segment to forward frame?
- Looks like a routing problem...

# Self learning

- A switch has a switch table
- entry in switch table:
  - (MAC Address, Interface, Time Stamp)
  - stale entries in table dropped (TTL can be 60 min)

- 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

# Filtering/Forwarding

When switch receives a frame:

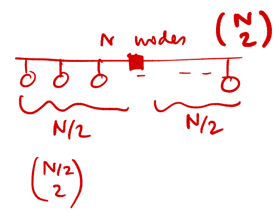
index switch table using MAC dest address

**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

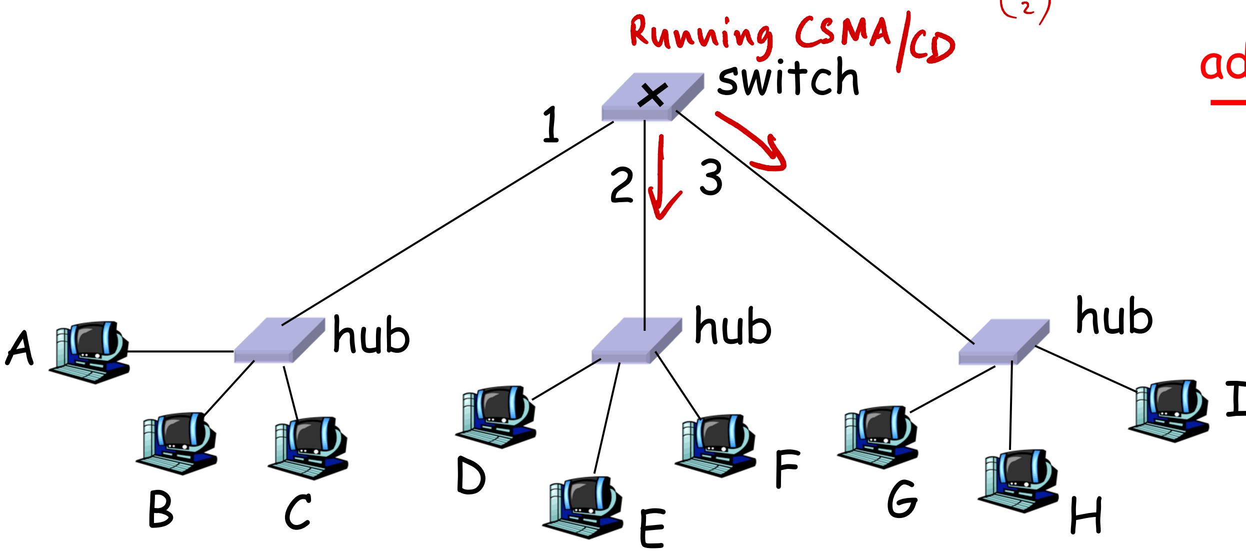
*forward on all but the interface on which the frame arrived*

# Switch example

Suppose C sends frame to D




$\binom{N}{2}$ vs $\binom{N/2}{2}$

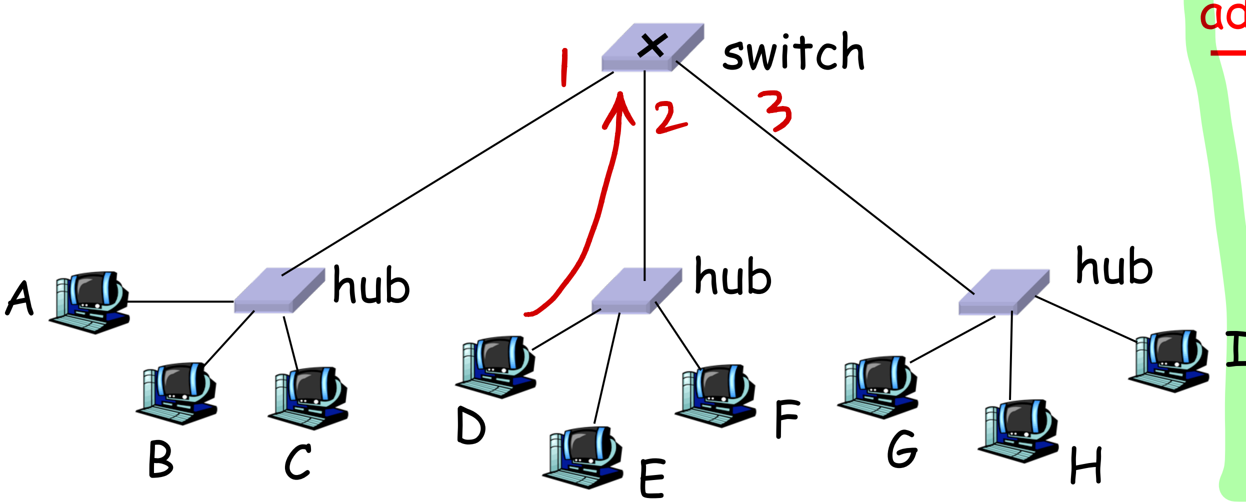



| address | interface |
|---|---|
| A | 1 |
| B | 1 |
| E | 2 |
| G | 3 |
| C | 1 |

- Switch receives frame from C
  - notes in bridge table that C is on interface 1
  - because D is not in table, switch forwards frame into interfaces 2 and 3
- frame received by D

# Switch example

Suppose D replies back with frame to C.



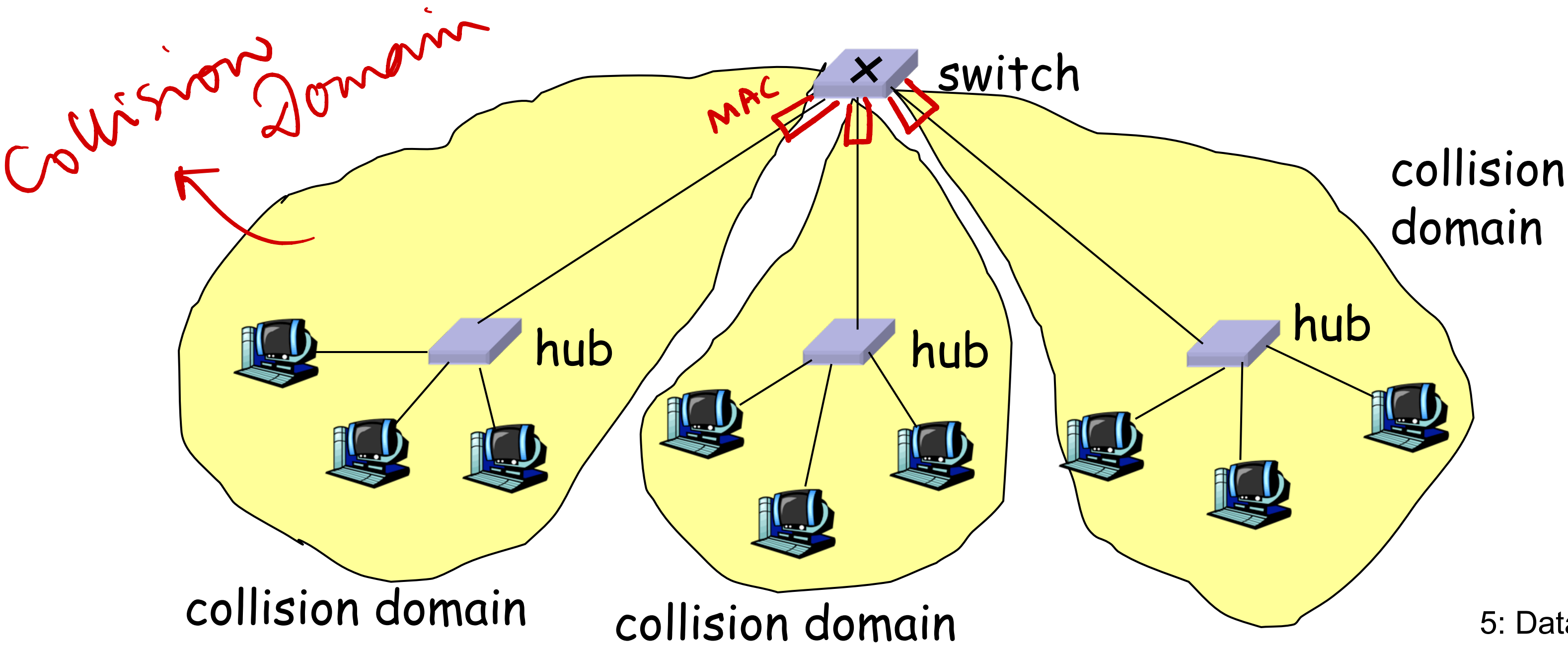

Switching table

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

- Switch receives frame from from D
  - notes in bridge table that D is on interface 2
  - because C is in table, switch forwards frame only to interface 1
- frame received by C

# Switch: traffic isolation

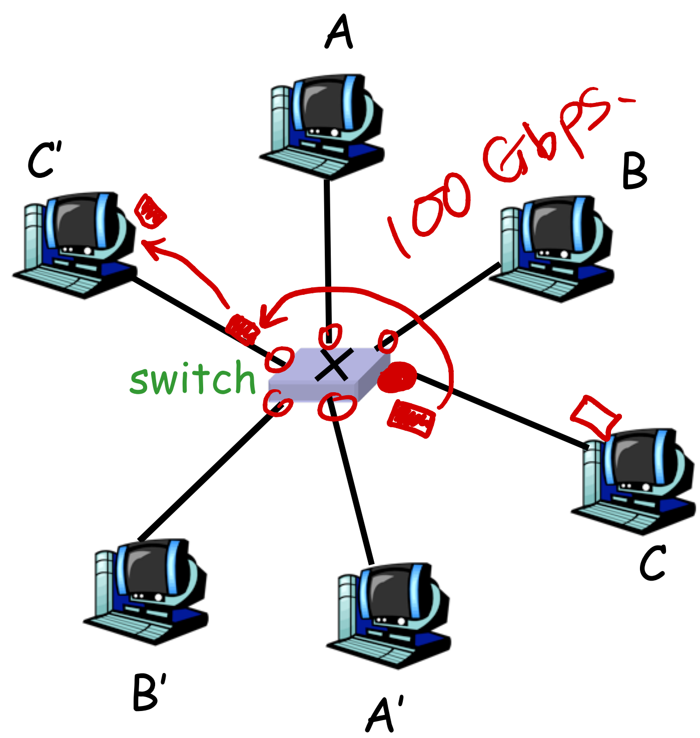
- switch installation breaks subnet into LAN segments
- switch filters packets:
  - same-LAN-segment frames not usually forwarded onto other LAN segments
  - segments become separate collision domains



# Switches: dedicated access

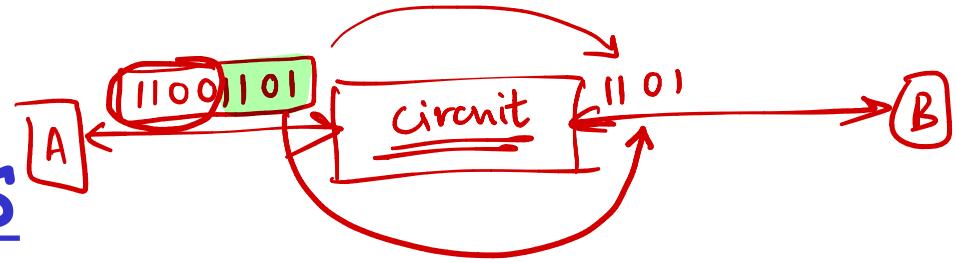
Latency

- Switch with many interfaces
- Hosts have direct connection to switch
- No collisions; full duplex




Switching: A-to-A' and B-to-B' simultaneously, no collisions

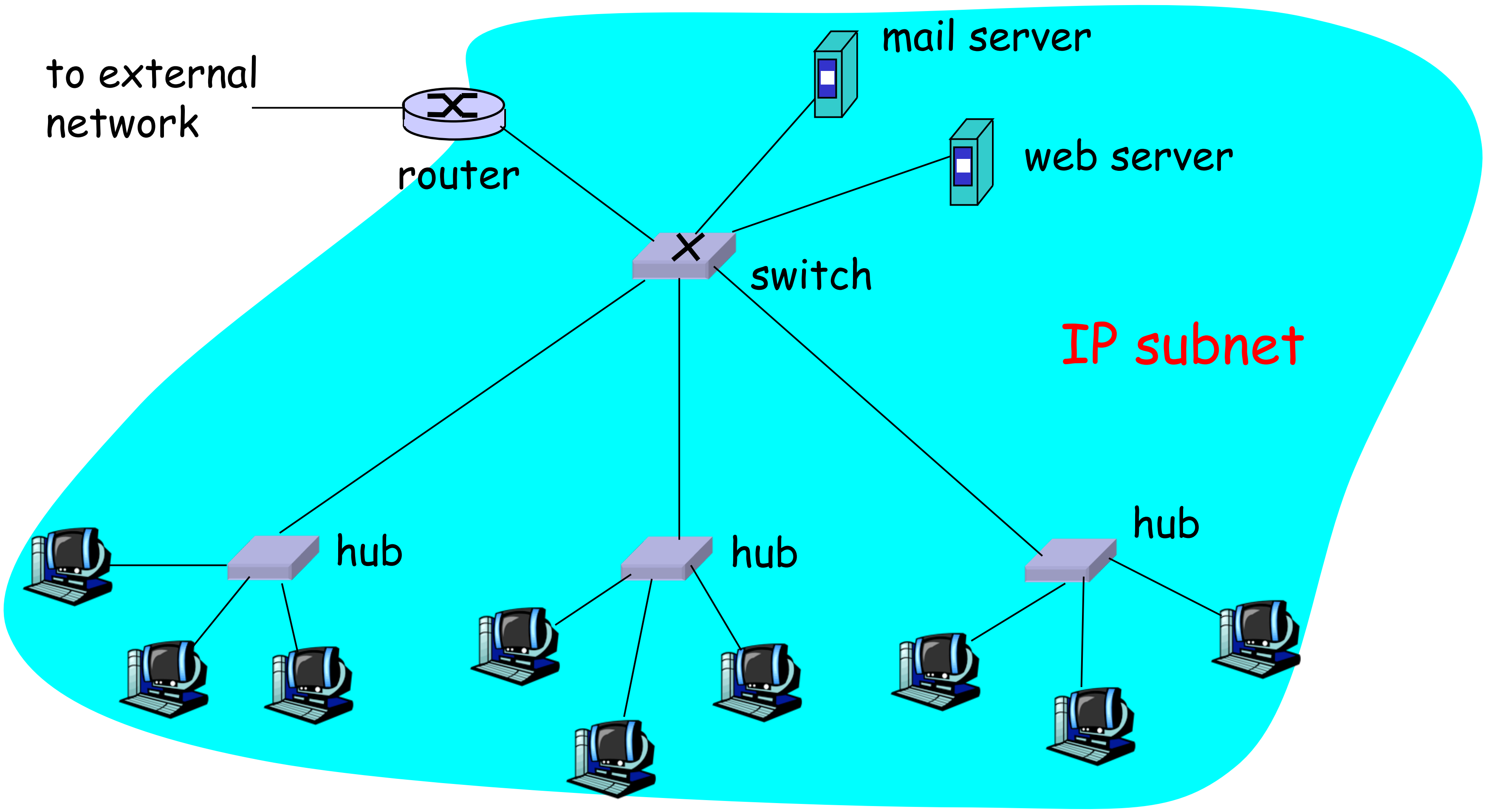
# More on Switches




- cut-through switching: frame forwarded from input to output port without first collecting entire frame
  - slight reduction in latency

- combinations of shared/dedicated, 10/100/1000 Mbps interfaces

# Institutional network

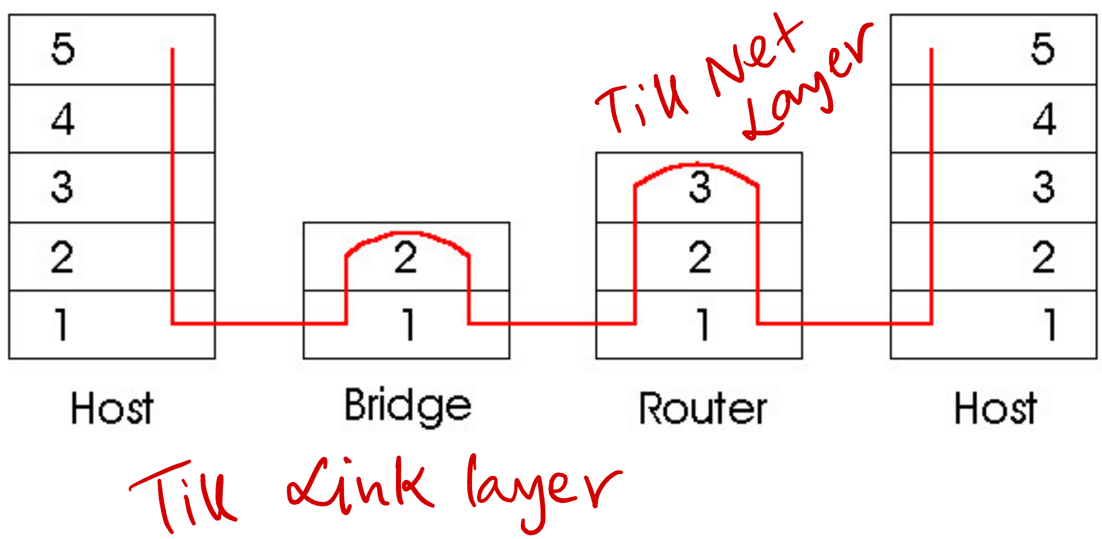


# So …

What's the difference between switches and routers?

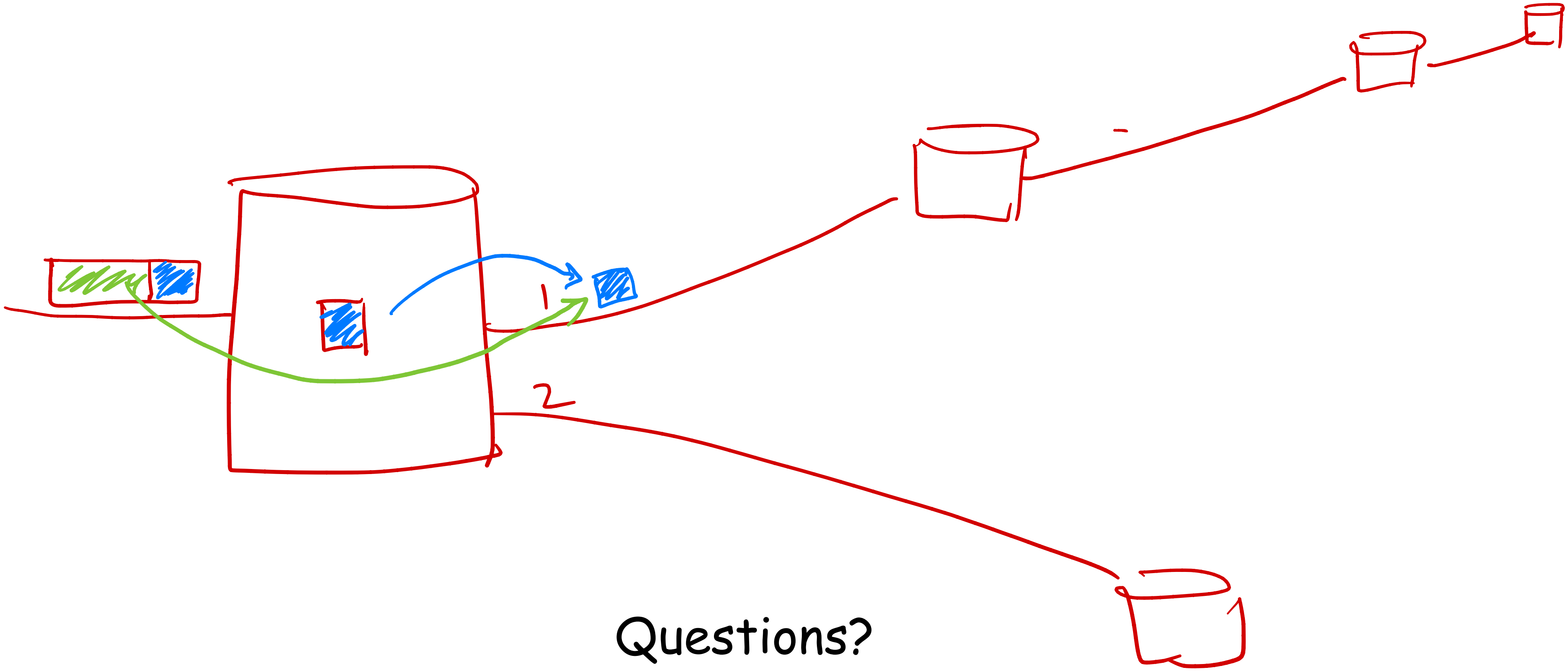
# Switches vs. Routers

- both store-and-forward devices
  - routers: network layer devices (examine network layer headers)
  - switches are link layer devices
- routers maintain routing tables, implement routing algorithms
- switches maintain switch tables, implement filtering, learning algorithms



# Summary comparison

| | hubs | routers | switches |
|---|---|---|---|
| traffic isolation | no | yes | yes |
| plug & play | yes | no | yes |
| optimal routing | no | yes | no |
| cut through | yes | no | yes |




Questions?