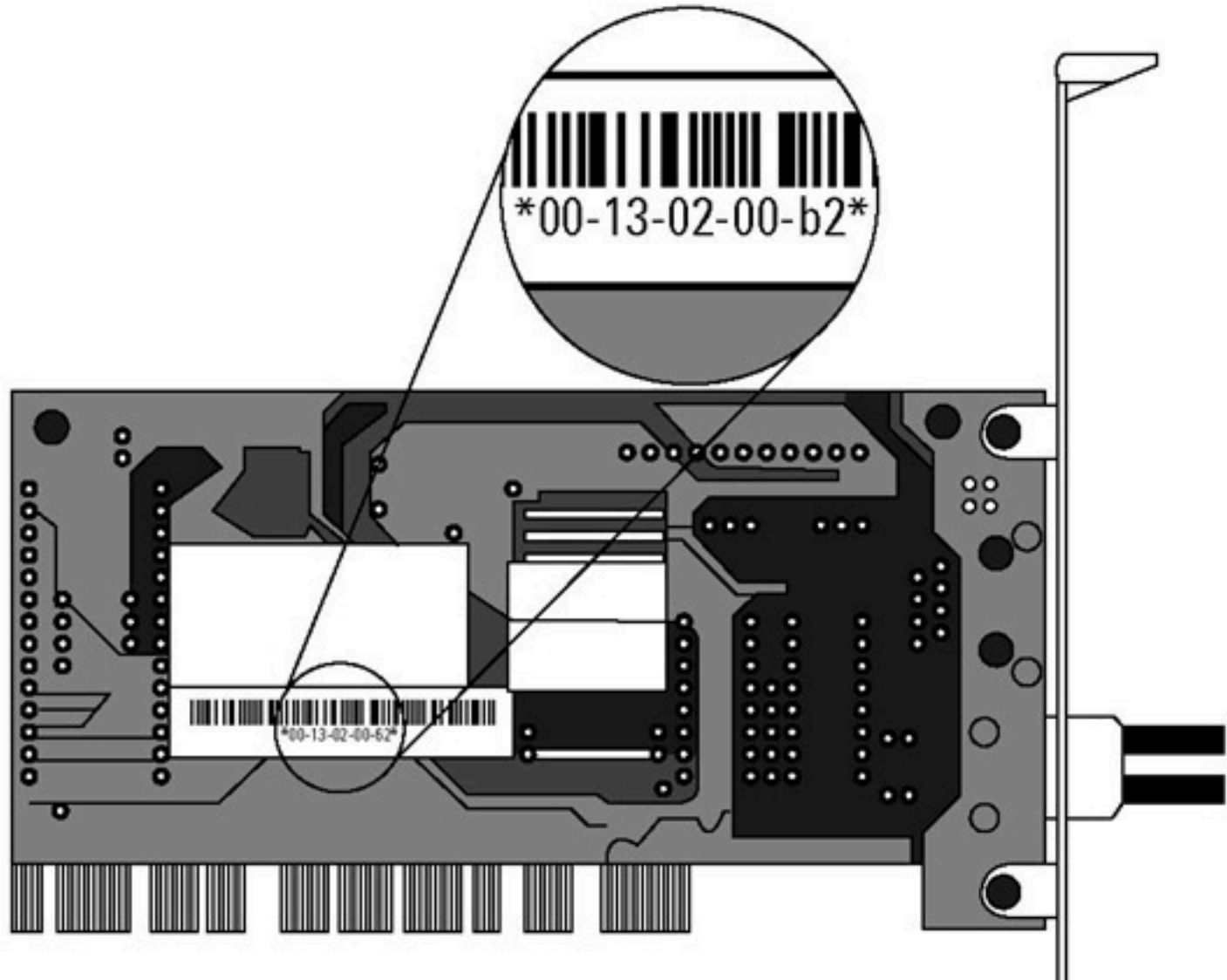


MAC addresses and ARP

- 32-bit IP address:
 - *network-layer* address for interface
 - used for layer 3 (network layer) forwarding
- MAC (or LAN or physical or Ethernet) address:
 - function: *used ‘locally’ to get frame from one interface to another physically-connected interface (same network, in IP-addressing sense)*
 - 48 bit MAC address (for most LANs) burned in NIC ROM (lately can be software settable)
 - e.g.: 1A-2F-BB-76-09-AD

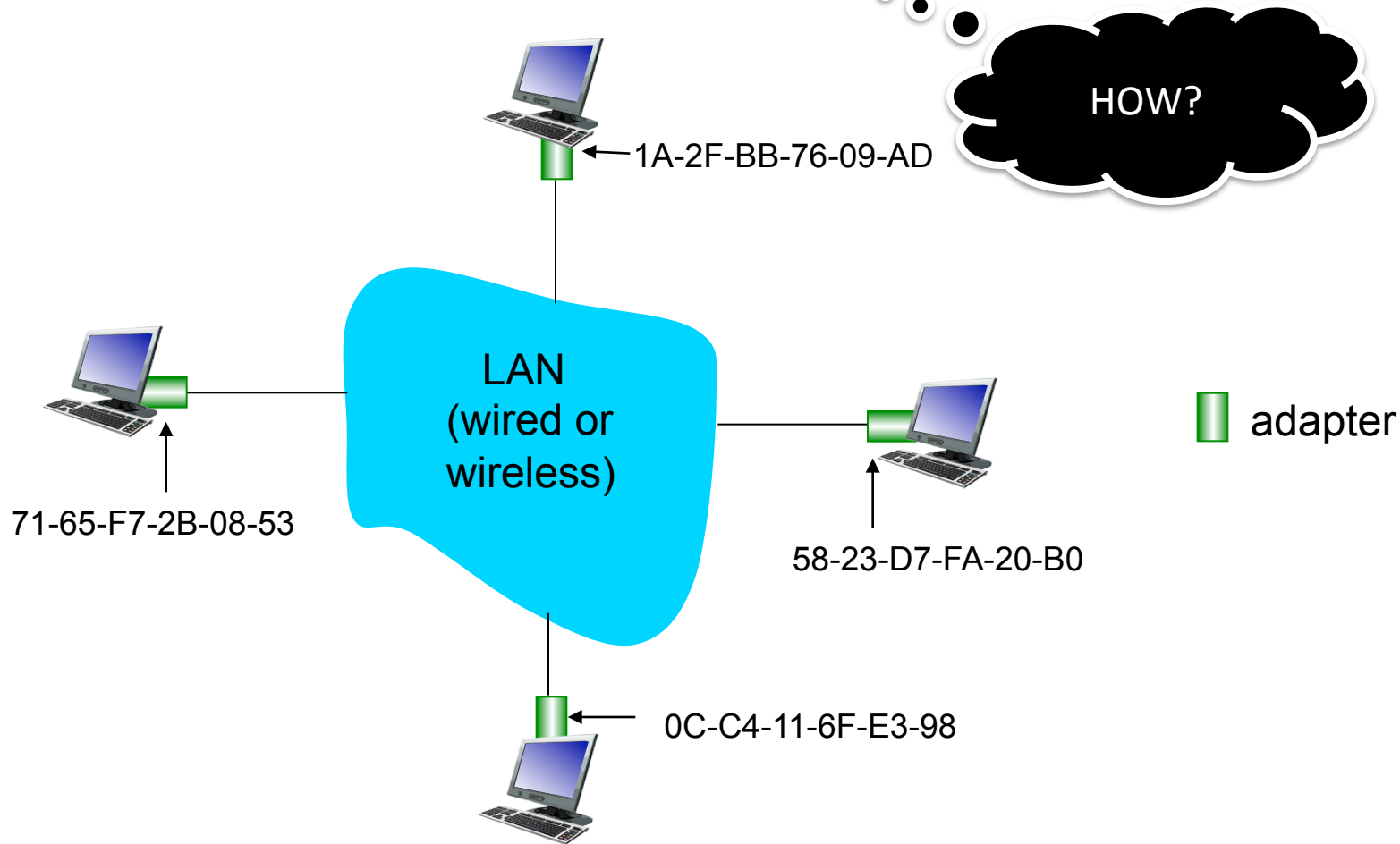
hexadecimal (base 16) notation
(each “number” represents 4 bits)

MAC address



LAN addresses and ARP

each adapter on LAN has unique **LAN** address

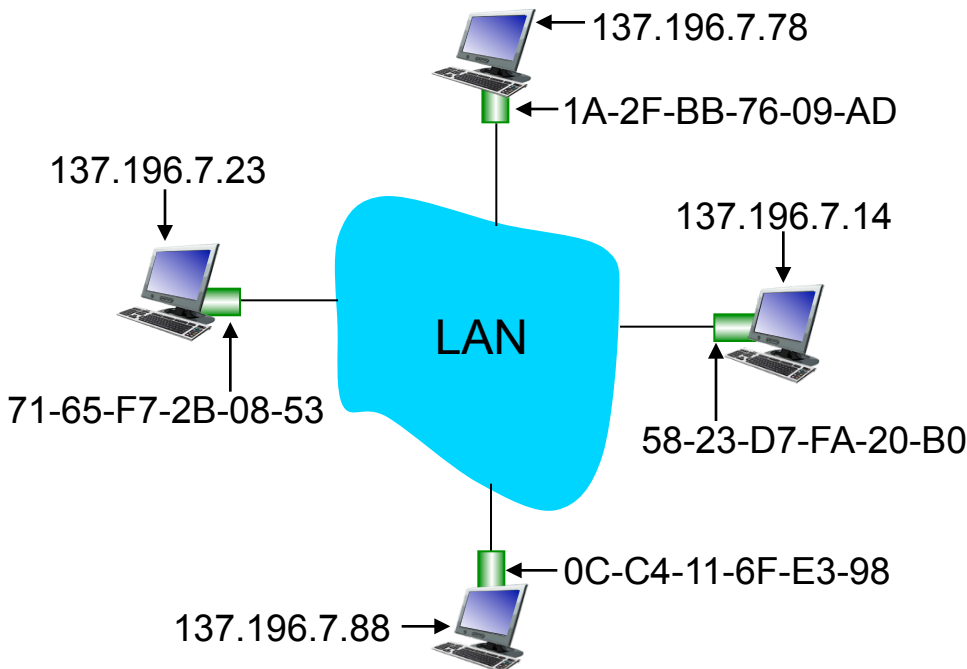


LAN addresses (more)

- 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 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: same LAN

- A wants to send datagram to B
 - 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 nodes 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*

ARP request:

who has 128.238.38.1 tell 128.238.38.160

```
▷ Frame 13: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)
▷ Ethernet II, Src: Ibm_10:60:99 (00:09:6b:10:60:99), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▽ Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IP (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: Ibm_10:60:99 (00:09:6b:10:60:99)
  Sender IP address: 128.238.38.160 (128.238.38.160)
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 128.238.38.1 (128.238.38.1)
```

ARP reply:

128.238.38.1 is at 00:00:0c:07:ac:00

```
▷ Frame 14: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
▷ Ethernet II, Src: All-HSRP-routers_00 (00:00:0c:07:ac:00), Dst: Ibm_10:60:99 (00:09:6b:10:60:99)
▽ Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IP (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: All-HSRP-routers_00 (00:00:0c:07:ac:00)
  Sender IP address: 128.238.38.1 (128.238.38.1)
  Target MAC address: Ibm_10:60:99 (00:09:6b:10:60:99)
  Target IP address: 128.238.38.160 (128.238.38.160)
```

ARP table

- A. Is formed and updated automatically by the ARP protocol
- B. Is used for same LAN
- C. Can be changed manually
- D. A and B
- E. A, B and C

Editing ARP table

The screenshot shows the 'ARP Table' utility window. The 'Adapters:' dropdown is set to '3Com EtherLink XL 10/100 PCI TX NIC (3C905B-TX)'. The 'ARP Table:' section shows 'Entries: 10 - [Static: 1 | Dynamic: 9]'. A table lists 10 entries with columns for '#', 'Type', 'IP Address', and 'MAC Address'. A 'Refresh' button is to the right of the table. Below the table are buttons for 'Add...', 'Edit...', and 'Remove'. An 'Add' dialog box is open, titled 'Enter IP and MAC Addresses', with fields for 'IP Address' (10 . 0 . 0 . 120) and 'MAC Address' (00 15 4D 22 12 09). The 'Static' checkbox is checked. 'OK' and 'Cancel' buttons are at the bottom of the dialog.

Adapters: 3Com EtherLink XL 10/100 PCI TX NIC (3C905B-TX)

ARP Table: Entries: 10 - [Static: 1 | Dynamic: 9]

#	Type	IP Address	MAC Address
1	Static	10.0.0.2	00-30-CD-F6-00-4E
2	Dynamic	10.0.0.6	00-E0-4E-20-37-87
3	Dynamic	10.0.0.8	00-1A-4D-6A-7F-48
4	Dynamic	10.0.0.11	00-E0-20-86-03-54
5	Dynamic	10.0.0.15	00-1D-7D-36-3D-65
6	Dynamic	10.0.0.25	00-15-58-1F-83-9F
7	Dynamic	10.0.0.28	00-15-58-3D-1B-73
8	Dynamic	10.0.0.61	
9	Dynamic	10.0.0.77	
10	Dynamic	10.0.0.113	

Refresh

Add...

Edit...

Remove

Enter IP and MAC Addresses

IP Address: 10 . 0 . 0 . 120

MAC Address: 00 15 4D 22 12 09

Static

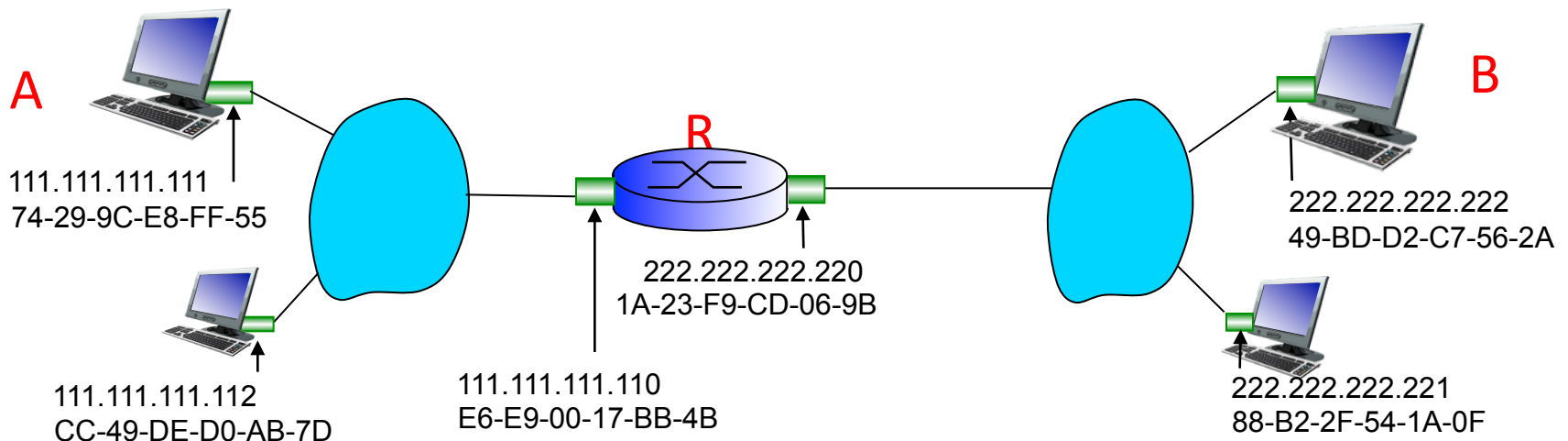
OK Cancel

By Eng. Usama El-Mokadem: musama@hotmail.com

Addressing: routing to another LAN

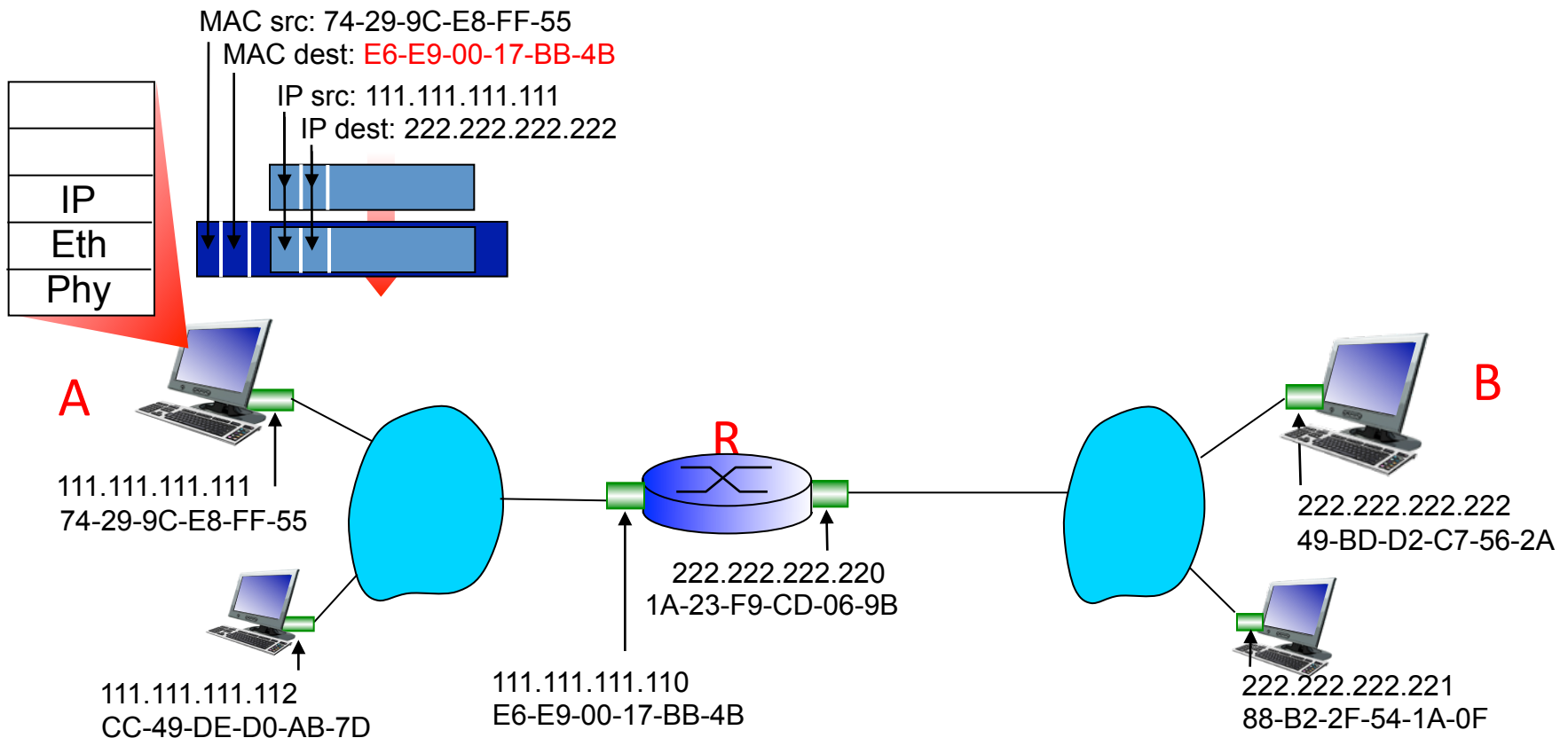
walkthrough: send datagram from A to B via R

- focus on addressing – at 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 R's MAC address (how?)



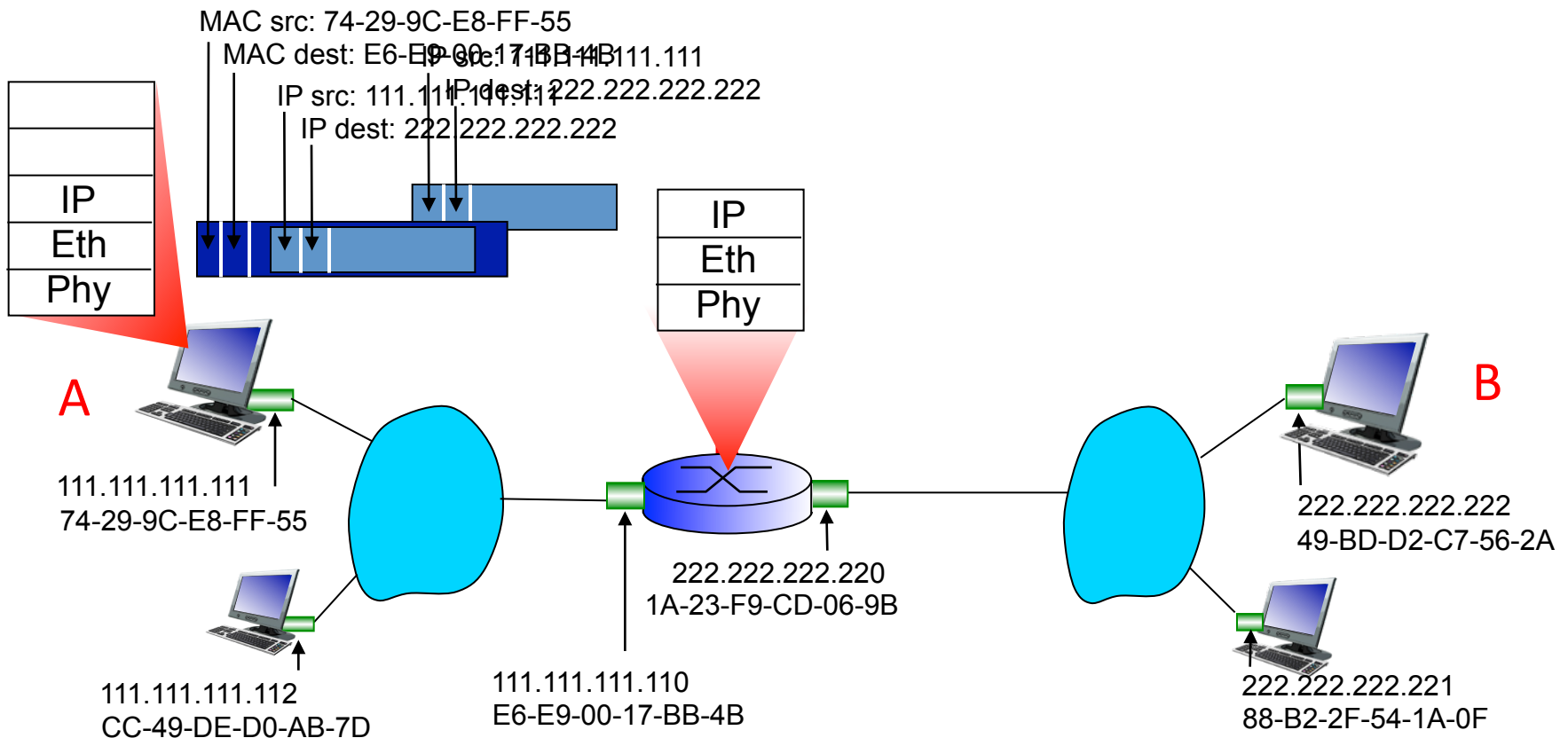
Addressing: routing to another LAN

- ❖ 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



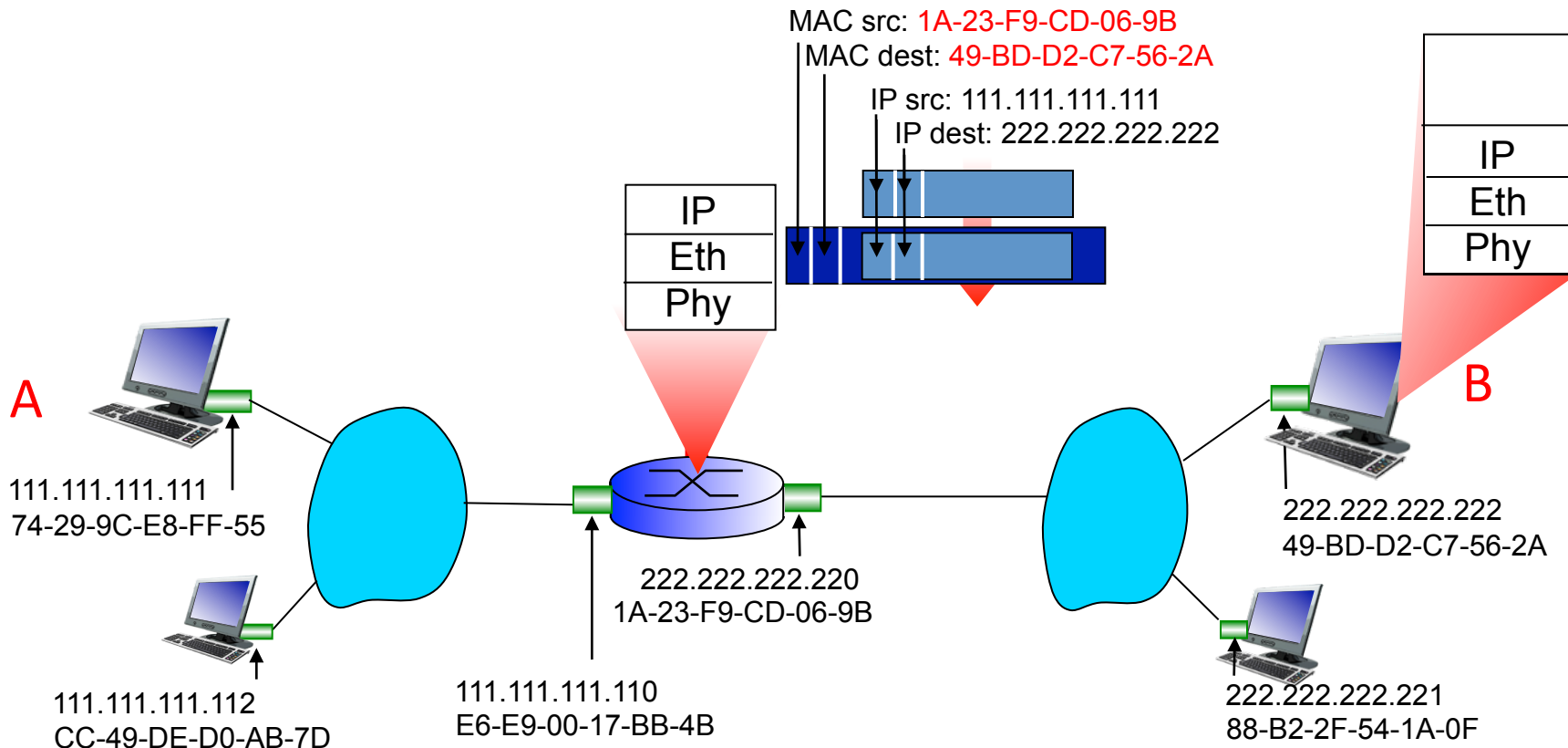
Addressing: routing to another LAN

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



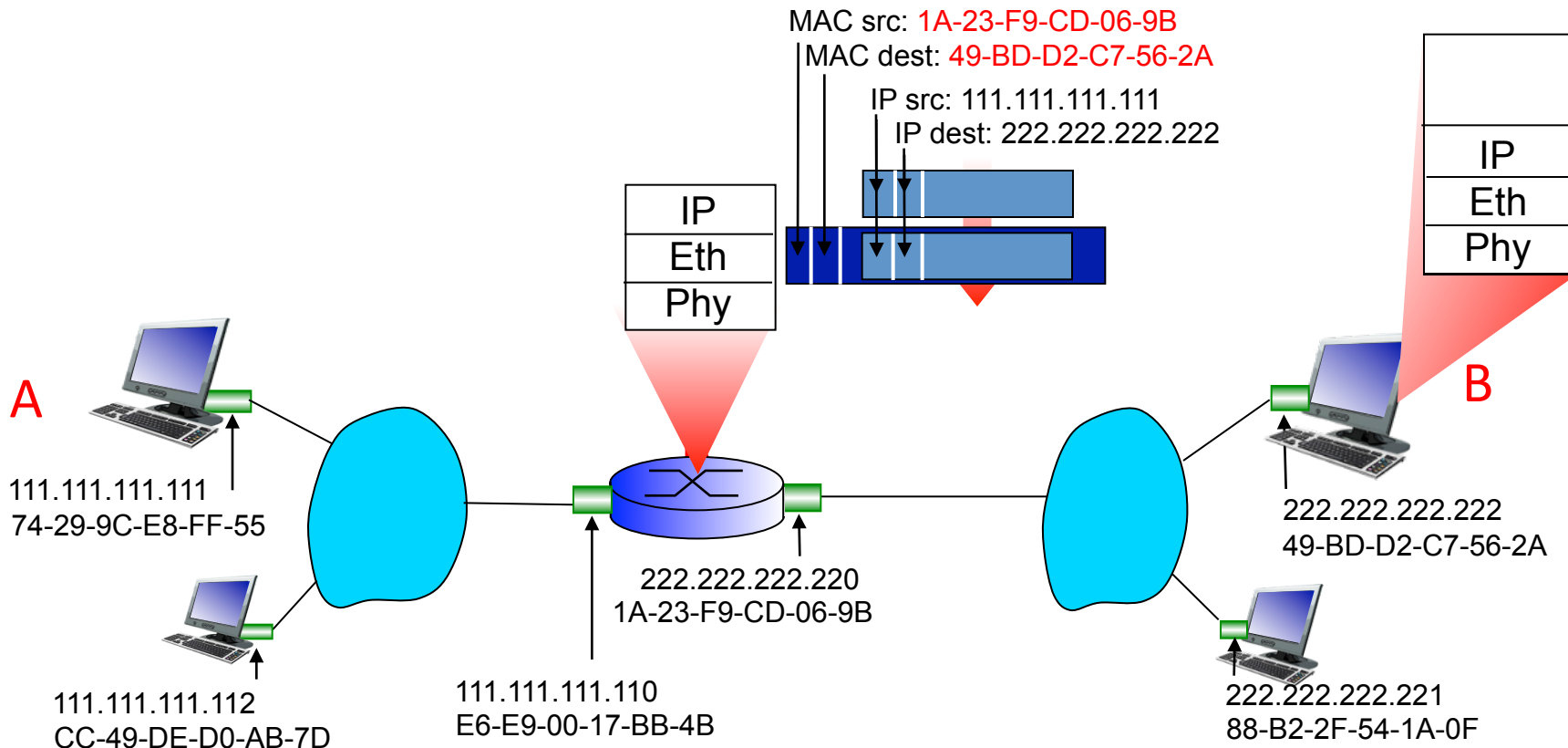
Addressing: routing to another LAN

- ❖ 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



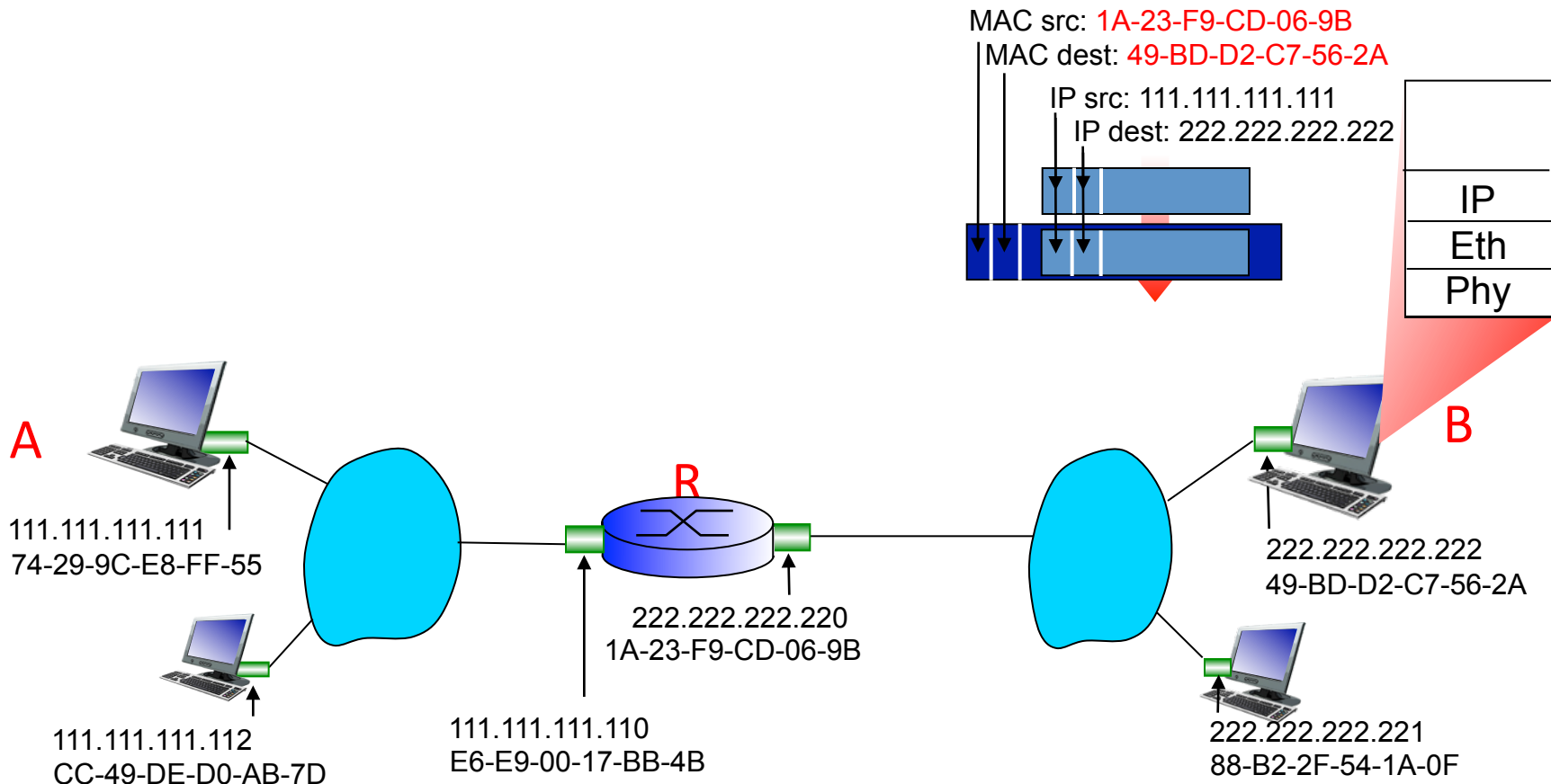
Addressing: routing to another LAN

- ❖ 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



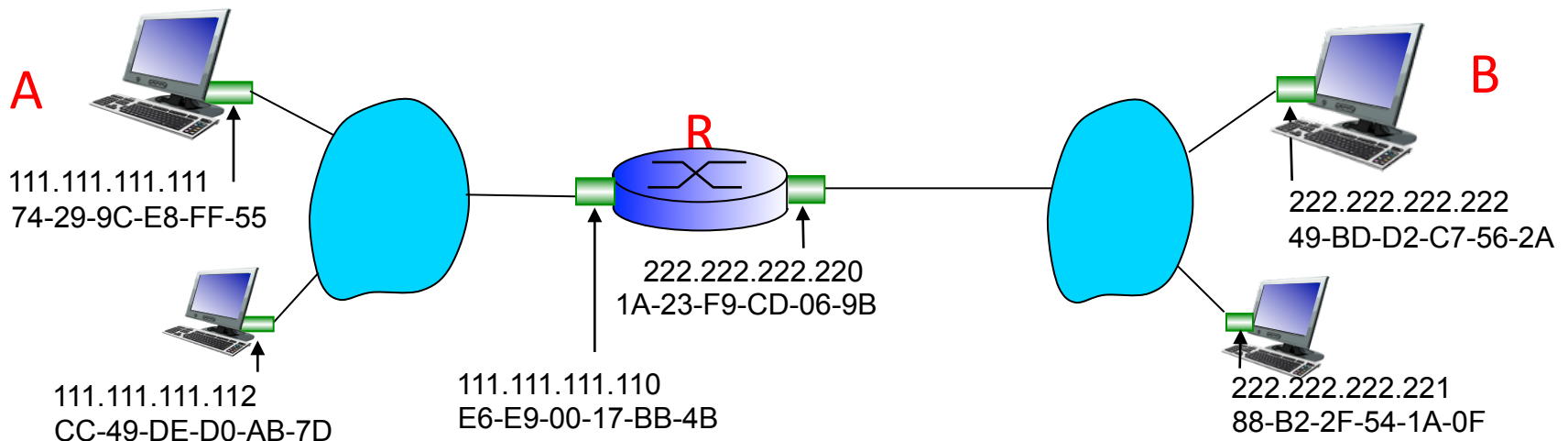
Addressing: routing to another LAN

- ❖ 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



How A knows IP address of R, MAC address of R

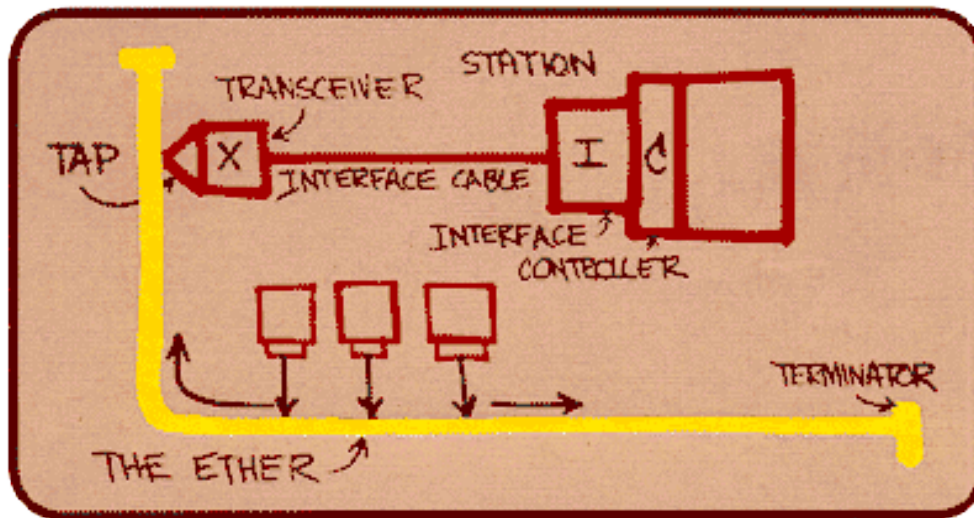
- A. DNS, DHCP
- B. IP, ARP
- C. DHCP, ARP
- D. DNS, IP



Ethernet

“dominant” wired LAN technology:

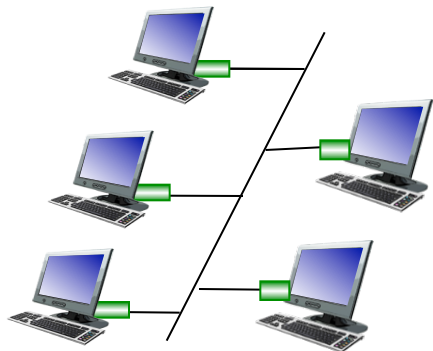
- cheap \$20 for NIC
- first widely used LAN technology
- simpler, cheaper than token LANs and ATM
- kept up with speed race: 10 Mbps – 10 Gbps



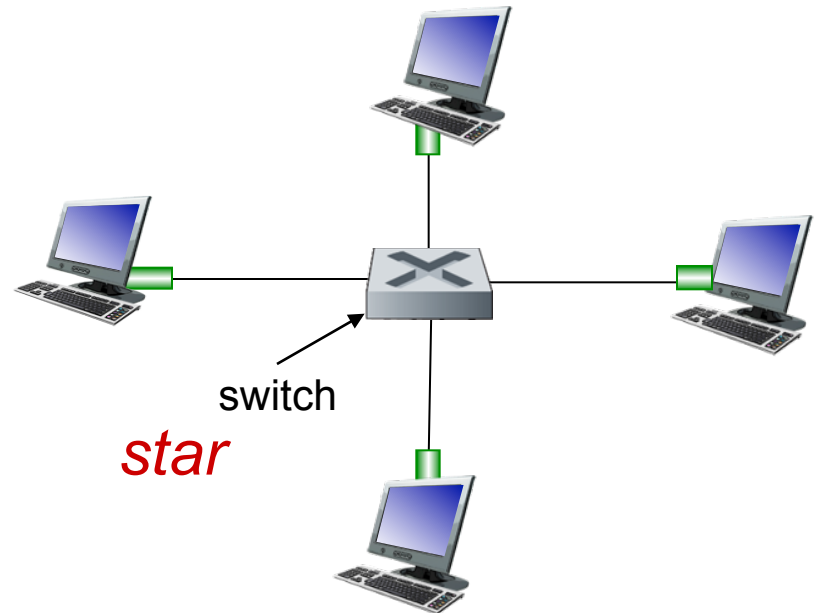
Metcalfe's Ethernet sketch

Ethernet: physical topology

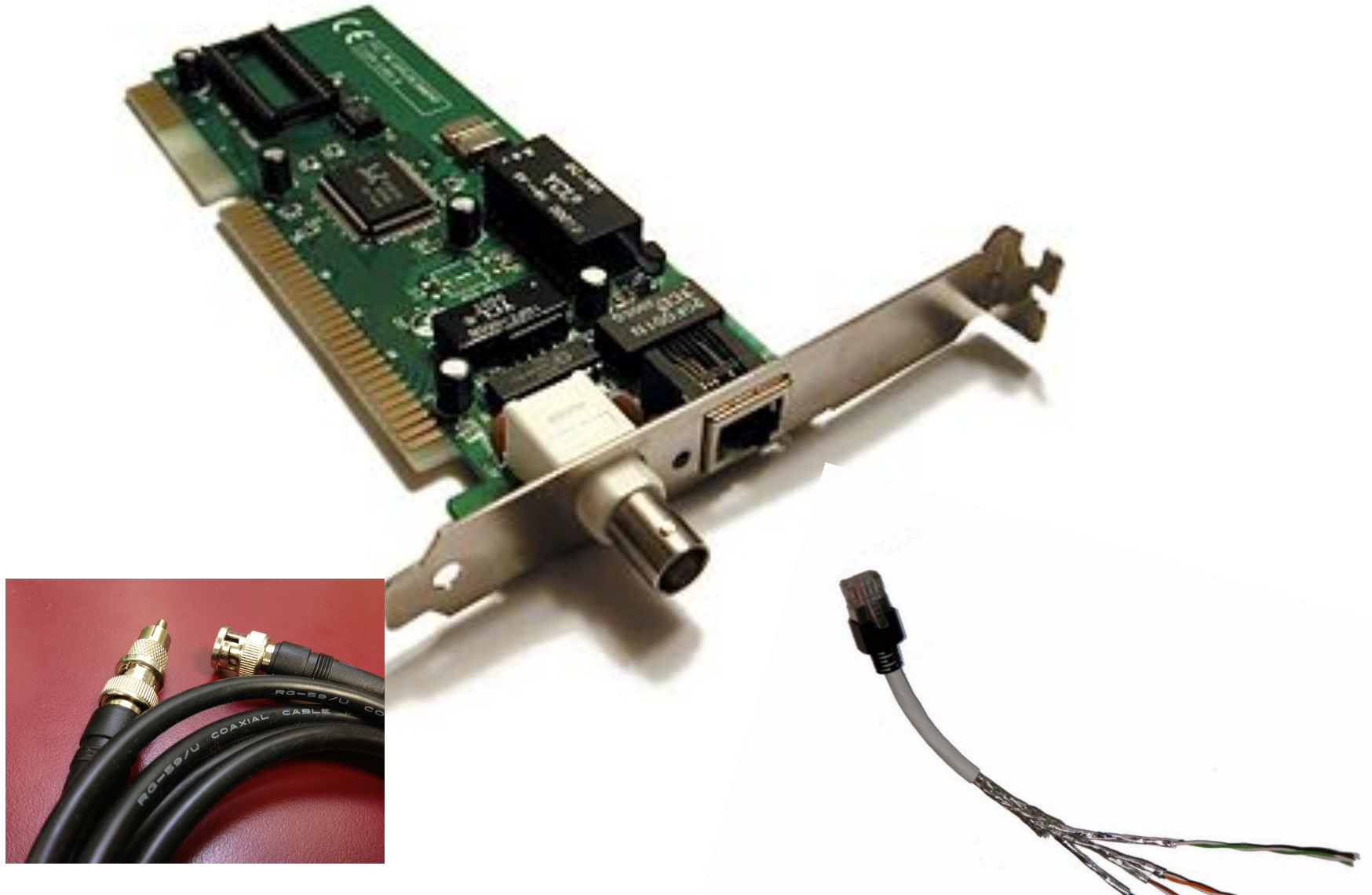
- **bus**: popular through mid 90s
 - all nodes in same collision domain (can collide with each other)
- **star**: prevails today
 - active **switch** in center
 - each “spoke” runs a (separate) Ethernet protocol (nodes do not collide with each other)



bus: coaxial cable



Coaxial cable and twister pair NIC



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

Preamble is used in Ethernet

- A. To synchronize receiver, sender clock rates
- B. To “wake up” the receiving adapter, i.e. to announce there will be frame coming
- C. To identify which protocol is used
- D. A and B
- E. A, B and C

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



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 rdt (e.g., TCP), otherwise dropped data lost
- Ethernet's MAC protocol: unslotted *CSMA/CD with binary backoff*

802.3 Ethernet standards: link & physical layers

- *many* different Ethernet standards
 - common MAC protocol and frame format
 - different speeds: 2 Mbps, 10 Mbps, 100 Mbps, 1 Gbps, 10G bps
 - different physical layer media: fiber, cable

