CS2105

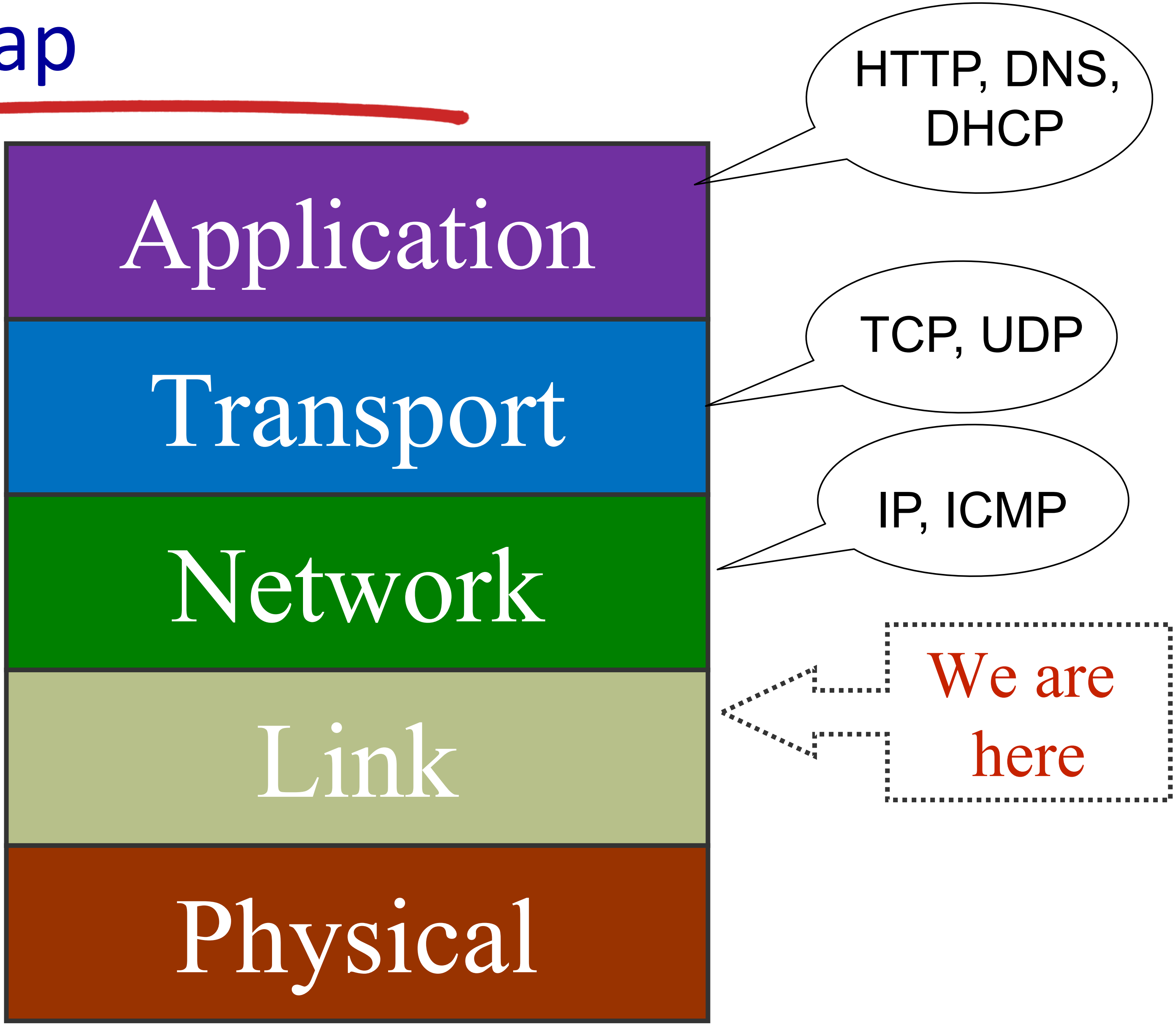
An *Awesome* Introduction to Computer Networks

The Link Layer, LAN

NUS National University of Singapore
Department of Computer Science
School of Computing

Adapted from Slides by Prof Roger
And
J.F Kurose and K.W. Ross, All Rights Reserved

Recap


Application
Transport
Network
Link
Physical
HTTP, DNS, DHCP
TCP, UDP
IP, ICMP
We are here

The Link Layer

After this set of lectures, we will understand:

- the role of link layer and the services it could provide.
- how parity and CRC scheme work.
- different methods for accessing shared medium.
- the role of switches in interconnecting subnets in a LAN.
- how ARP allows a host to discover the MAC addresses of other nodes in the same subnet.

Roadmap

6.1 Introduction to the Link Layer

6.2 Error Detection and Correction

6.3 Multiple Access Links and Protocols

6.4 Switched Local Area Networks

- 6.4.1 Link Layer Addressing & ARP
- 6.4.2 Ethernet
- 6.4.3 Link-layer Switches

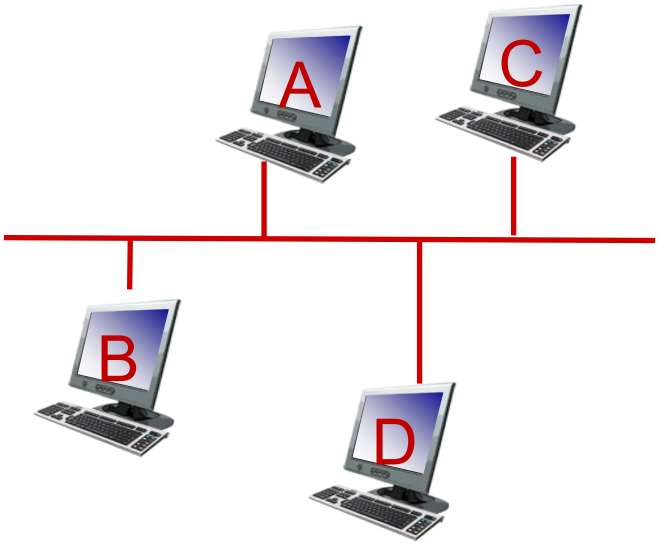
Motivation

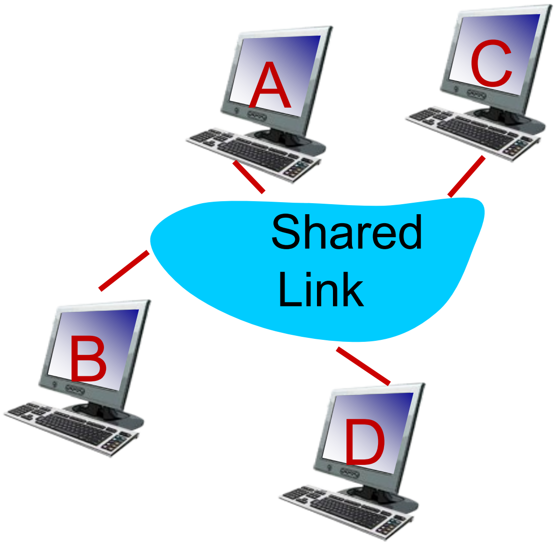
- *Aim:* Send data between *N* nodes via cable.
- *Solution:* Inter-Connect the *N* nodes via a broadcast link
  - Each link needs to be *addressed*
  - Need to define a *protocol*
  - Need to handle *errors*

Detection

Link Access Control

Framing

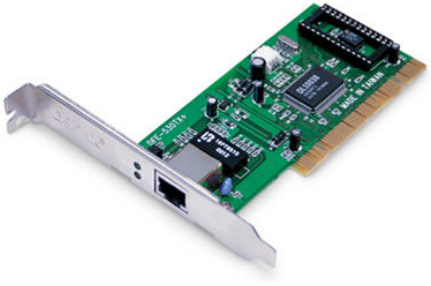




MAC Address

Jargon Alert:

- MAC: *Media Access Control*



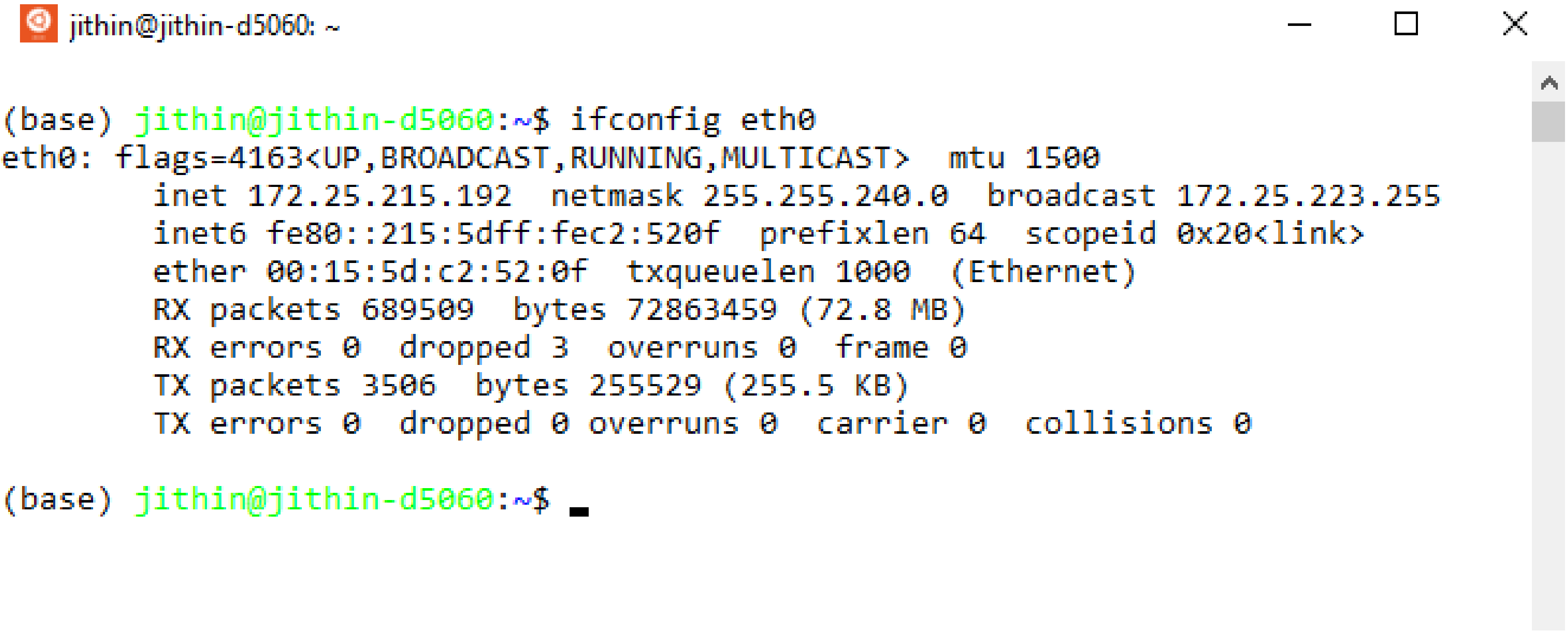
- Every adapter (NIC) has a MAC address (aka physical or LAN address).
  - Used to send and receive link layer frames.
  - When an adapter receives a frame, it checks if the destination MAC address of the frame matches its own MAC address.
    - If yes, adapter extracts the enclosed datagram and passes it to the protocol stack.
    - If no, adapter simply discards the frame without interrupting the host.

MAC Address

Jargon Alert:
- NIC: *Network Interface Card*
- ROM: *Read-only Memory*

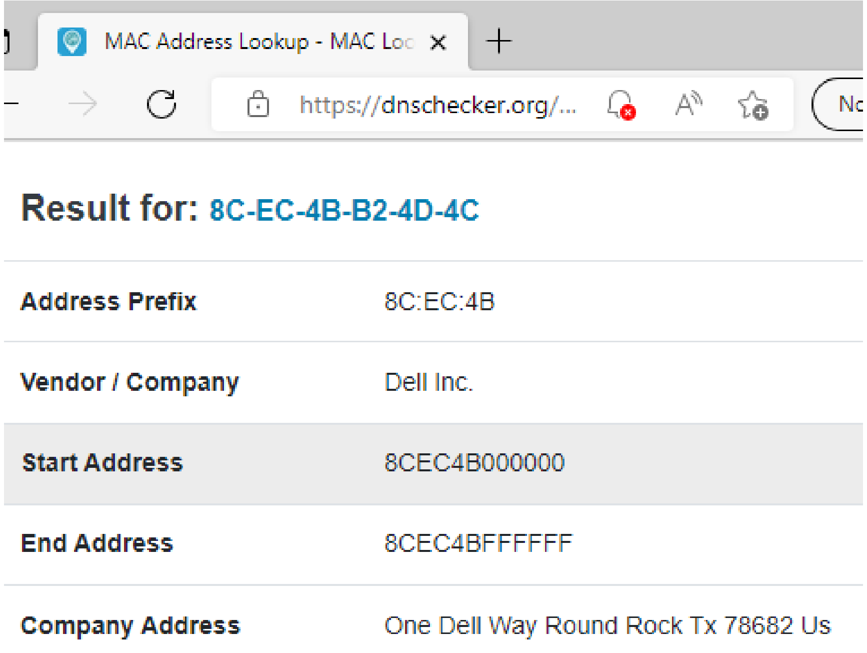
- MAC address is typically 48 bits, burned in NIC ROM (sometimes software settable).
  - Example: **5C-F9-DD-E8-E3-D2** — hexadecimal (base 16) notation
    - 0101 1100 1111 1001 1101 1101 1110 1000 1110 0011 1101 0010
  - MAC address allocation is administered by IEEE.
    - The first three bytes identifies the vendor of an adapter.
  - Broadcast Address: **FF-FF-FF-FF-FF-FF**

MAC Address



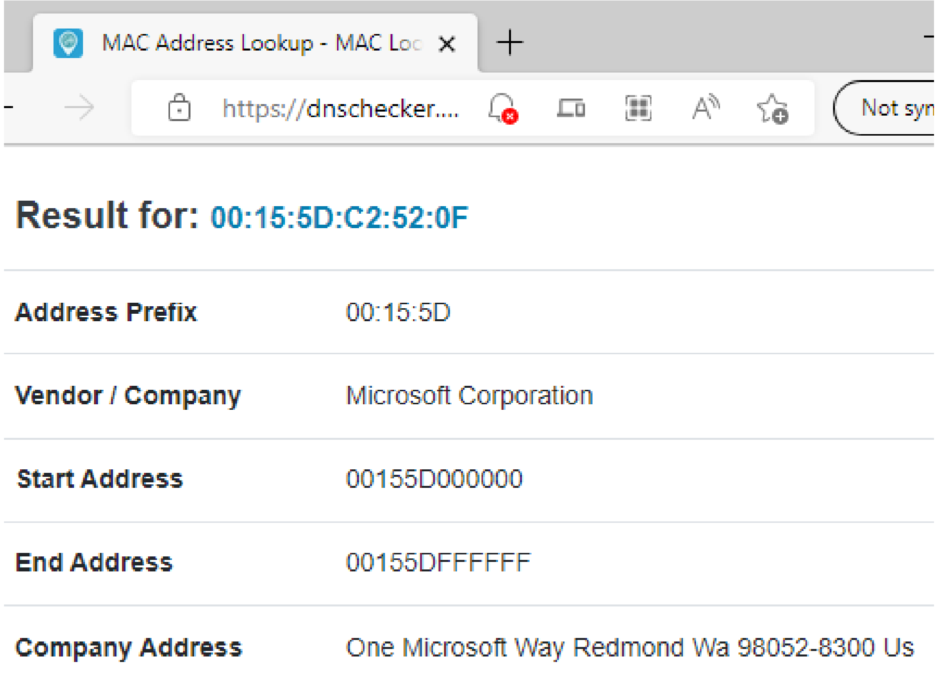
```
(base) jithin@jithin-d5060:~$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.25.215.192  netmask 255.255.240.0  broadcast 172.25.223.255
        inet6 fe80::215:5dff:fec2:520f  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:c2:52:0f  txqueuelen 1000  (Ethernet)
        RX packets 689509  bytes 72863459 (72.8 MB)
        RX errors 0  dropped 3  overruns 0  frame 0
        TX packets 3506  bytes 255529 (255.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

(base) jithin@jithin-d5060:~$ _
```

MAC Address



Result for: 8C-EC-4B-B2-4D-4C

Address Prefix	8C:EC:4B
Vendor / Company	Dell Inc.
Start Address	8CEC4B000000
End Address	8CEC4BFFFFFF
Company Address	One Dell Way Round Rock Tx 78682 Us

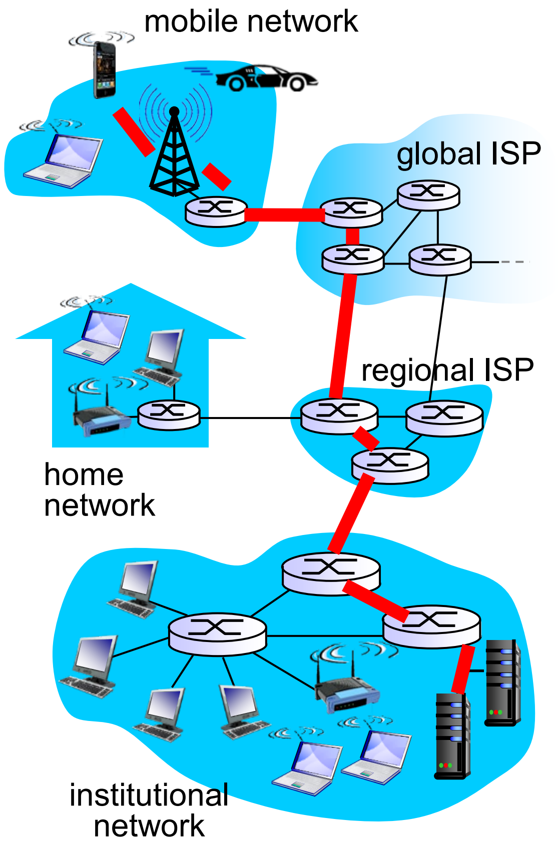


Result for: 00:15:5D:C2:52:0F

Address Prefix	00:15:5D
Vendor / Company	Microsoft Corporation
Start Address	00155D000000
End Address	00155DFFFFFF
Company Address	One Microsoft Way Redmond Wa 98052-8300 Us

Link Layer: Introduction

- *Link layer* sends datagram between adjacent nodes (hosts or routers) over a single link.
  - IP datagrams are encapsulated in link-layer frames for transmission.
  - Different link-layer protocols may be used on different links.
    - *each protocol may provide a different set of services.*



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

Roadmap

6.1 Introduction to the Link Layer

6.2 Error Detection and Correction

6.3 Multiple Access Links and Protocols

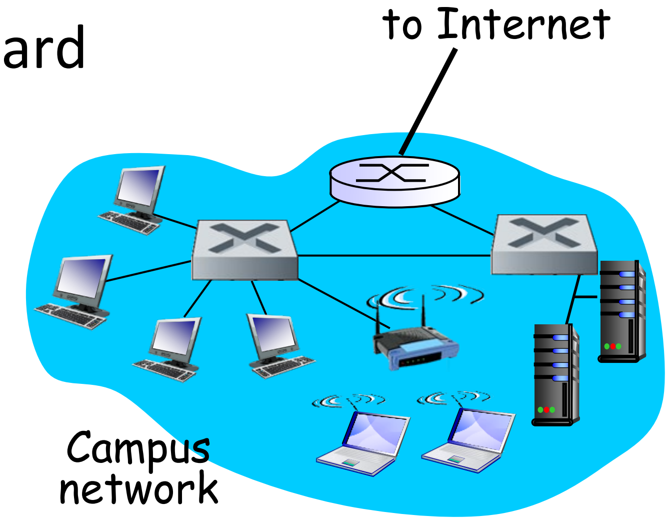
6.4 Switched Local Area Networks

- 6.4.1 Link Layer Addressing & ARP
- 6.4.2 Ethernet
- 6.4.3 Link-layer Switches

Local Area Network (LAN)

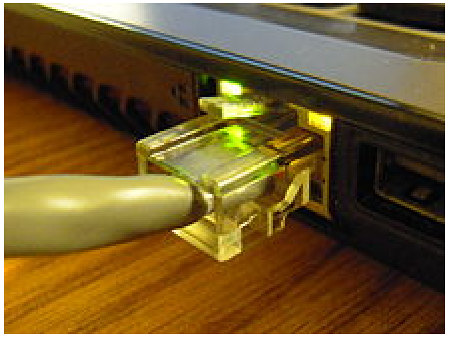
- LAN is a computer network that interconnects computers within a *geographical area* such as office building or university campus.

- LAN technologies:
  - IBM Token Ring: IEEE 802.5 standard
  - Ethernet: IEEE 802.3 standard
  - Wi-Fi: IEEE 802.11 standard
  - Others

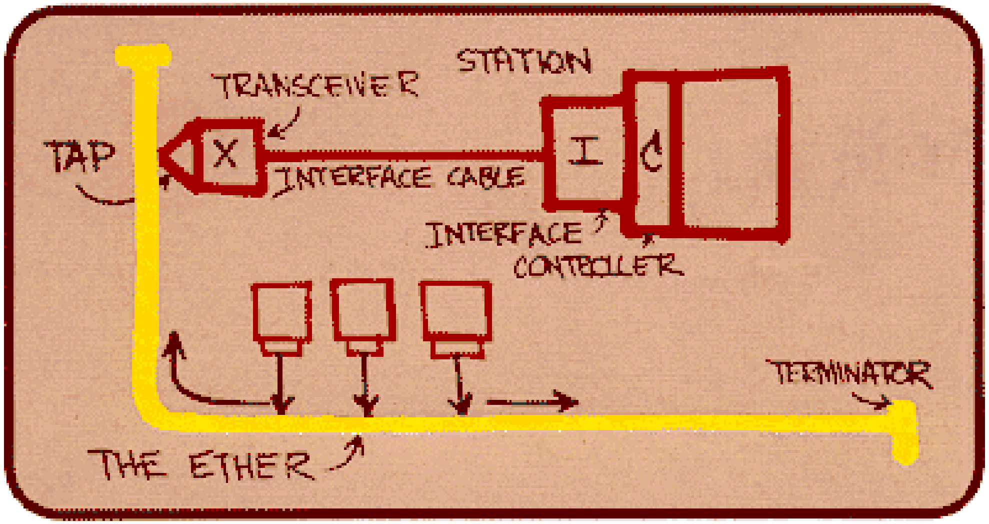


Ethernet

- "*Dominant*" wired LAN technology:
  - Developed in mid 1970s
  - Standardized by Xerox, DEC, and Intel in 1978
  - Simpler and cheaper than token ring and ATM



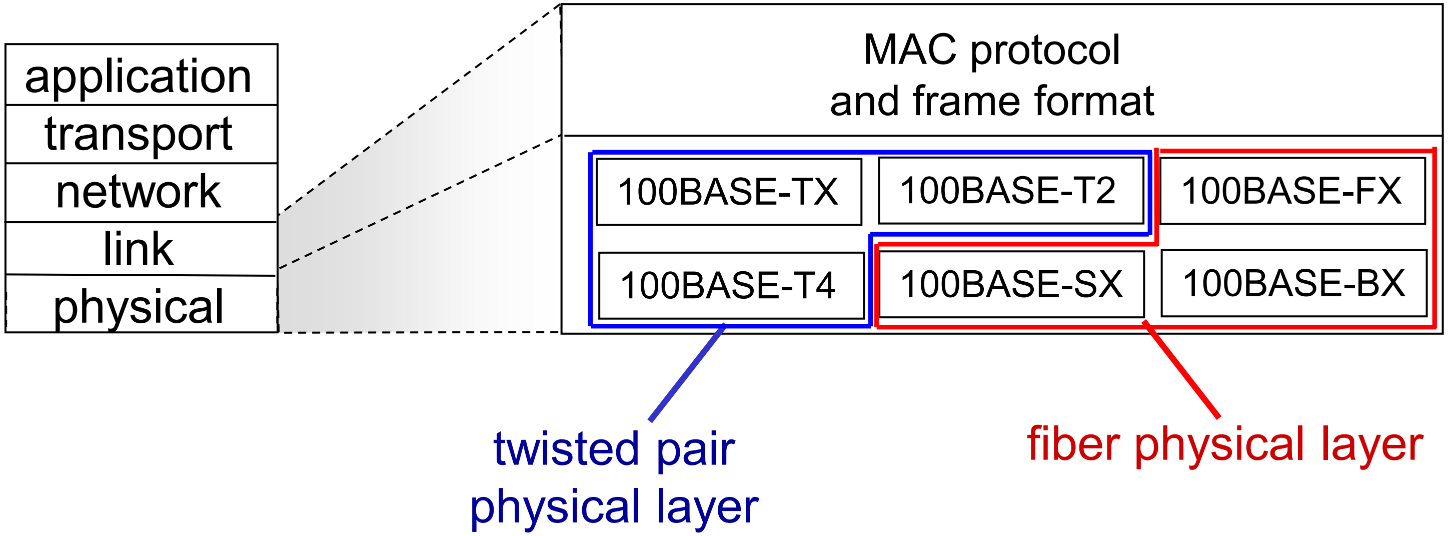
Ethernet connection
(Source: Wikipedia)



Metcalfe's Ethernet sketch

802.3 Ethernet Standards

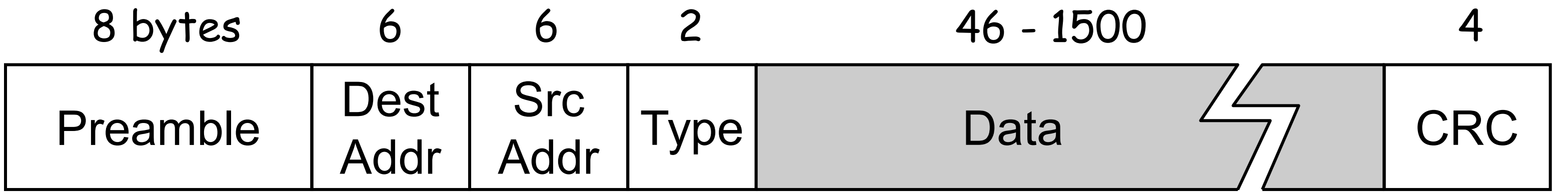
- A series of Ethernet standards have been developed over the years.
  - Different speeds: 2 Mbps, 10 Mbps, 100 Mbps, 1 Gbps, 10 Gbps, 100 Gbps
  - Different physical layer media: cable, fiber optics
  - MAC protocol and frame format remain unchanged



Ethernet Frame Structure

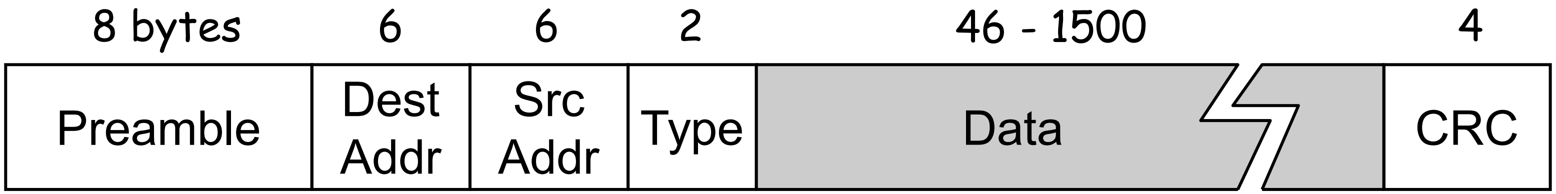
- Let us consider the case of sending an IP datagram from one host to another, on the same Ethernet LAN

- Sending NIC (adapter) encapsulates IP datagram in Ethernet frame.



8 bytes	6	6	2	46 - 1500	4
Preamble	Dest Addr	Src Addr	Type	Data	CRC

Ethernet Frame Structure



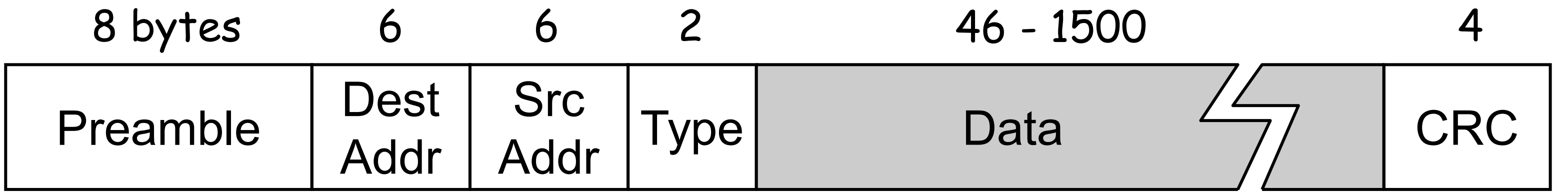
8 bytes	6	6	2	46 - 1500	4
Preamble	Dest Addr	Src Addr	Type	Data	CRC

- *Source and dest MAC address:*
  - If NIC receives a frame with matching destination address, or with broadcast address
    - it passes data in the frame to network layer protocol.
  - Otherwise
    - NIC discards frame.

Ethernet Frame Structure

Jargon Alert:

- MTU: *Maximum Transmission Unit*



8 bytes	6	6	2	46 - 1500	4
Preamble	Dest Addr	Src Addr	Type	Data	CRC

❖ *Data:*

- The maximum size is 1500 bytes.
  - This maximum size is the link MTU which we mentioned when we discussed IP fragmentation.
- The minimum size is 46 bytes
  - The minimum size is to ensure that a collision will always be detected.

Ethernet Frame Structure

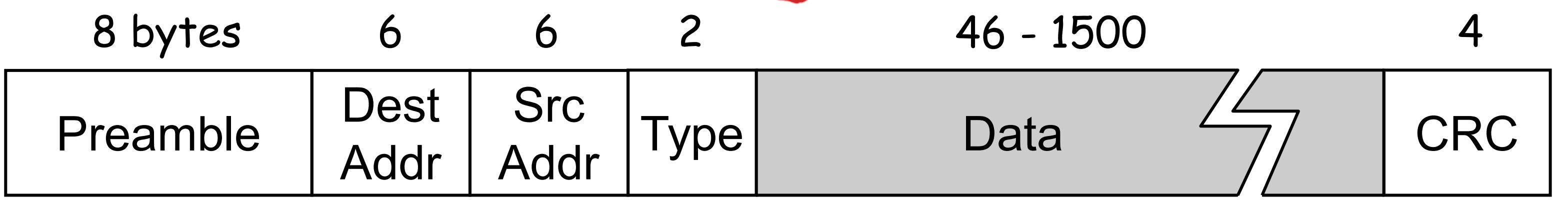
Jargon Alert:

- CRC: *Cyclic Redundancy Check*

8 bytes	6	6	2	46 - 1500	4
Preamble	Dest Addr	Src Addr	Type	Data	CRC

- *CRC:* corrupted frame will be dropped.
- *Type:* Indicates higher layer protocol
  - To understand this, we need to keep in mind that hosts can use other network-layer protocols besides IP.
    - E.g. Novell IPX, AppleTalk, ARP, etc.
  - The type field permits Ethernet to multiplex network-layer protocols.
  - Type field is analogous to
    - the protocol field in the network-layer datagram and
    - the port-number fields in the transport-layer segment

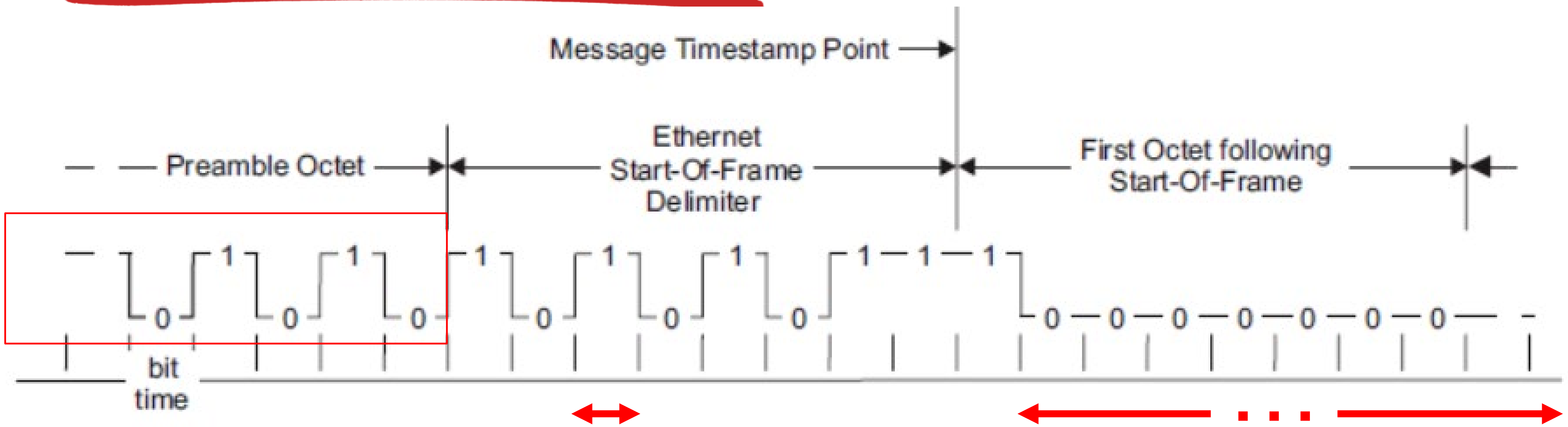
Ethernet Frame Structure



8 bytes	6	6	2	46 - 1500	4
Preamble	Dest Addr	Src Addr	Type	Data	CRC

- *Preamble:*
  - 7 bytes with pattern **10101010** (AA$_{\text{Hex}}$)
  - Followed by 1 byte with pattern **10101011** (AB$_{\text{Hex}}$).
    - *Also called "start of frame"*
  - used to synchronize receiver and sender clock rates.

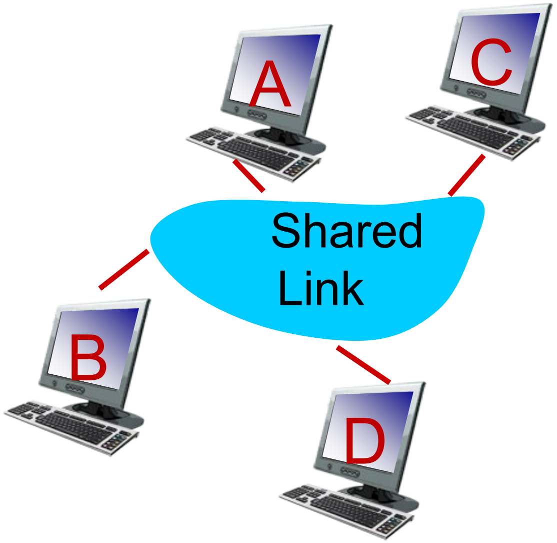
Ethernet Frame Structure


Message Timestamp Point
Preamble Octet
Ethernet Start-Of-Frame Delimiter
First Octet following Start-Of-Frame
1
0
bit time

19 or 20?

- The preamble provides a “square wave” pattern that tells the receiver the sender’s clock rate
  - it tells the receiver the width of a bit
  - which is important if there is a long string of bits of the same value, e.g., 19 or 20 zeros.

Motivation (revisited)



- *Aim:* Send data between *N* nodes via cable.
- *Solution:* Inter-Connect the *N* nodes via a broadcast link
  - Each link needs to be *addressed*
  - Need to define a *protocol*
  - Need to handle *errors*

Framing

Link Access Control

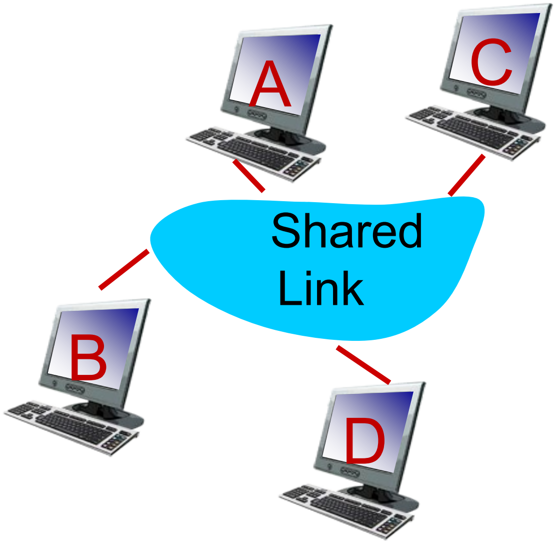
Reliability

Detection

Ethernet Data Delivery Service

- *Unreliable*: receiving NIC doesn't send ACK or NAK to sending NIC.
  - data in dropped frames will be recovered only if initial sender uses higher layer rdt (e.g. TCP); otherwise dropped data is lost.
- Ethernet's multiple access protocol:
  - *CSMA/CD* with binary (exponential) backoff.

Motivation (revisited)



- *Aim:* Send data between *N* nodes via cable.
- *Solution:* Inter-Connect the *N* nodes via a broadcast link
  - Each link needs to be *addressed*
  - Need to define a *protocol*
  - Need to handle *errors*

Framing

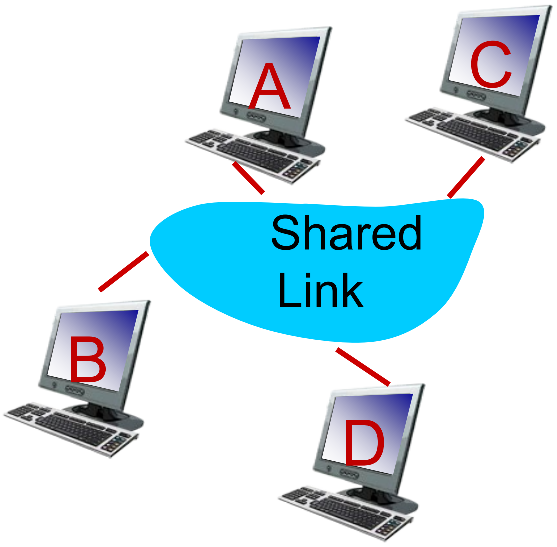
Link Access Control

Reliability

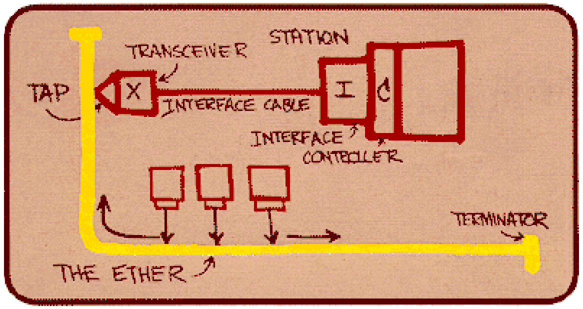
Detection

Ethernet: Physical Topology

- How do we interconnect the nodes to create this shared link?

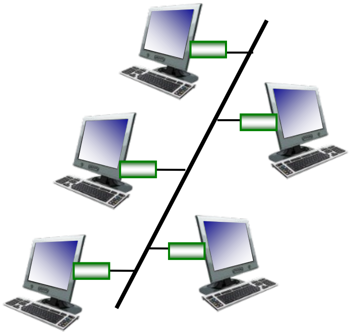


Ethernet: Bus Topology



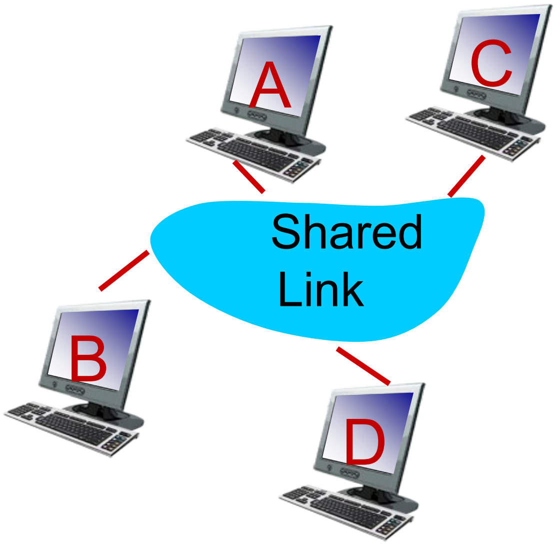
Bus topology: popular till mid 90s

- The *original* Ethernet LAN used a coaxial bus to interconnect the nodes.
- Is a *broadcast* LAN
  - All transmitted frames received by all adapters connected to the bus.
    - *all nodes can collide with each other*
- Drawbacks
  - Back bone cable
    - *If damaged, the entire network will fail*
  - Difficult to troubleshoot problems
  - Very slow and not ideal for larger networks
    - Due to collisions



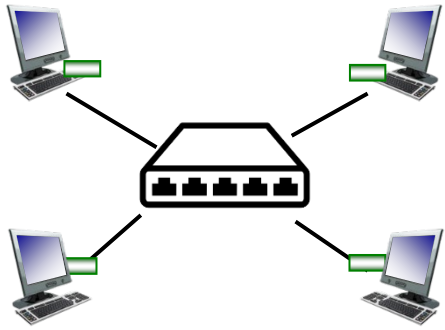
Ethernet with bus topology

Ethernet: Star Topology



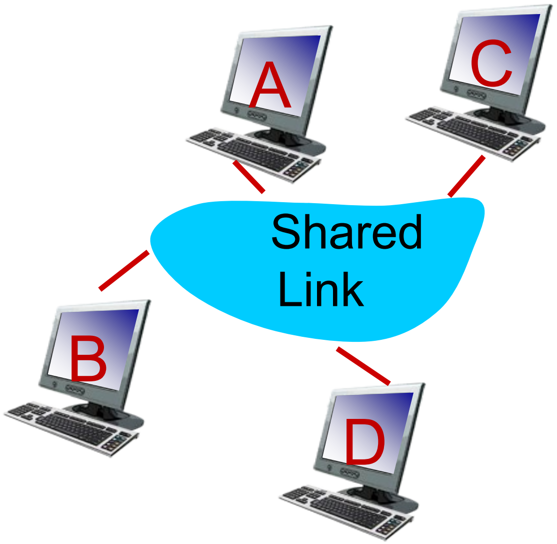
Star topology: prevalent today

- Hub
  - Popular in late 1990's
  - nodes are directly connected to a hub
  - A hub is a *physical-layer* device that acts on individual bits rather than frames.
    - When a bit arrives from one interface,
      - the hub simply re-creates the bit
      - boosts its energy strength, and
      - transmits the bit onto all the other interfaces.



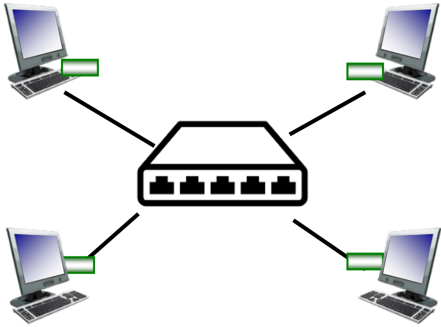
Ethernet with star topology

Ethernet: Star Topology



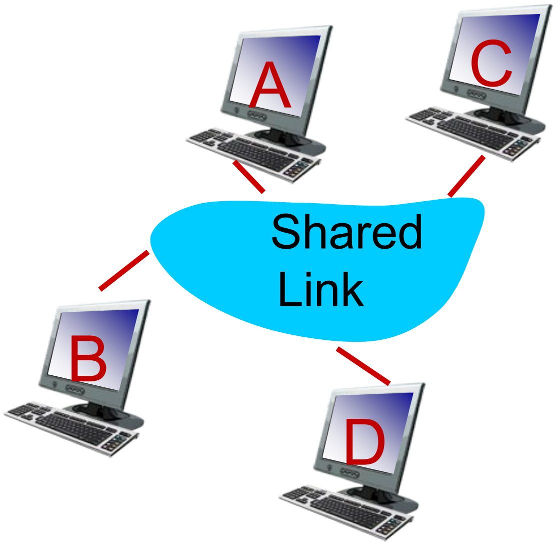
Star topology: prevalent today

- *Hub*
  - Advantages
    - Cheap
    - Easy Maintenance
      - Modular design of the network
  - Drawbacks
    - Very slow and not ideal for larger networks
      - Due to collisions



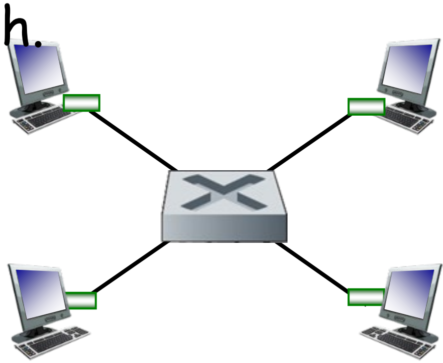
Ethernet with star topology

Ethernet: Star Topology



Star topology: prevalent today

- *Switch*
  - Popular since early 2000's
  - nodes are directly connected to a switch
  - A switch is a *layer-2* device
    - Works acts on *frames* rather than individual bits.
    - *No collisions*
    - A bona-fide *store-and-forward* packet switch.



Ethernet with star topology

Roadmap

6.1 Introduction to the Link Layer

6.2 Error Detection and Correction

6.3 Multiple Access Links and Protocols

6.4 Switched Local Area Networks

- 6.4.1 Link Layer Addressing & ARP
- 6.4.2 Ethernet
- 6.4.3 Link-layer Switches

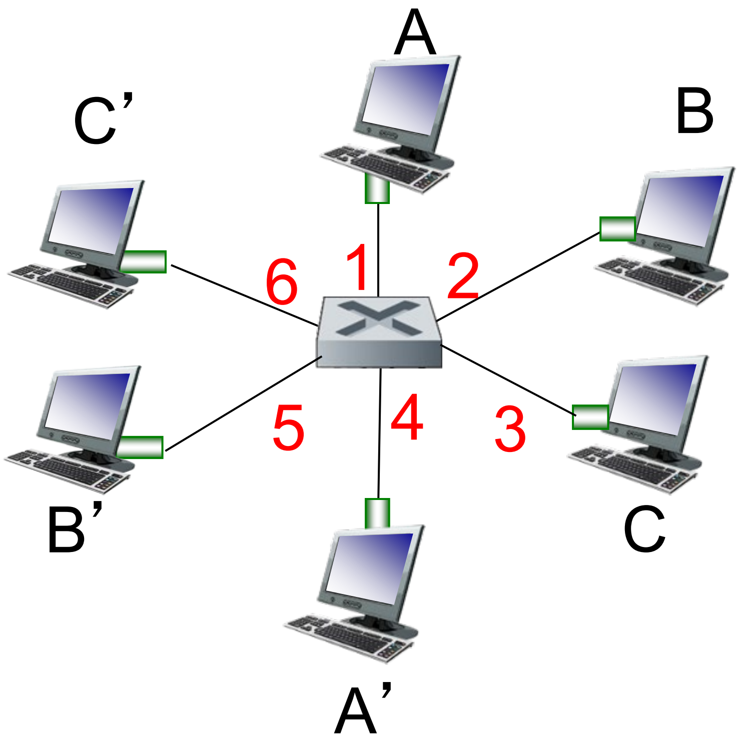
Ethernet Switch

a 50-port Ethernet switch (Source: Wikipedia)

- A *link-layer device* used in LAN
  - Examine incoming frame's MAC address
    - *selectively forward* frame to one-or-more outgoing links.
  - *Store and forward* Ethernet frames
  - uses *CSMA/CD* to access link
- *Transparent*
  - hosts are unaware of presence of switches
- *Plug-and-play (self-learning)*
  - switches do not need to be configured

Switch: *multiple* simultaneous transmissions

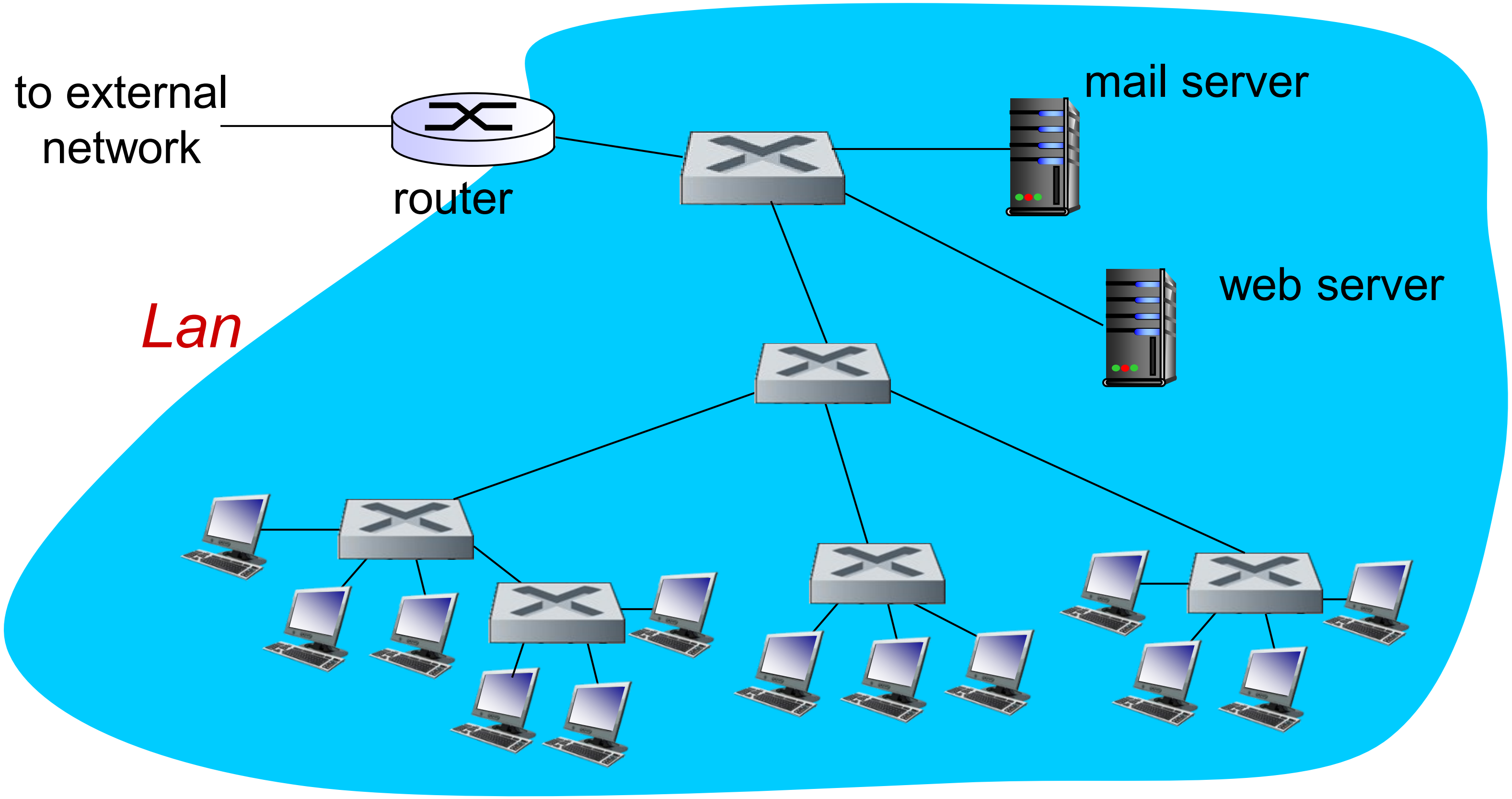
- Nodes have *dedicated*, *direct* connection to switch
- switches *buffer* packets
- Ethernet protocol used on *each* incoming link
  - but *no* collisions!
- *Switching:*
  - A-to-A' and B-to-B' can transmit simultaneously, without collisions



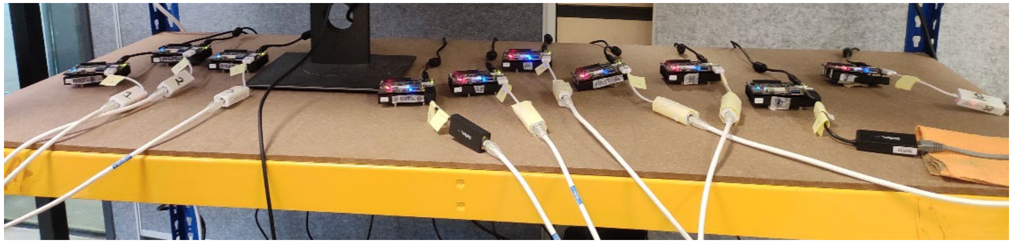
switch with six interfaces (1,2,3,4,5,6)

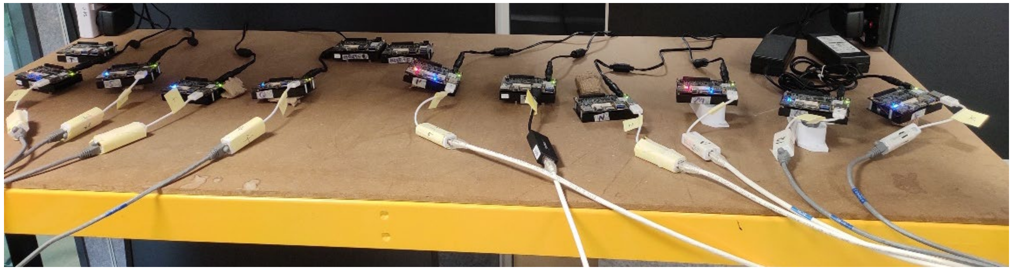
Interconnecting Switches

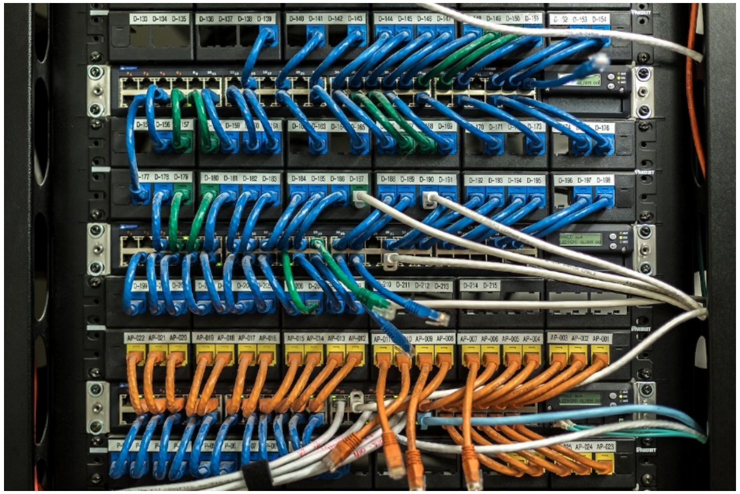
- Switches can be connected in hierarchy.



Interconnecting Switches





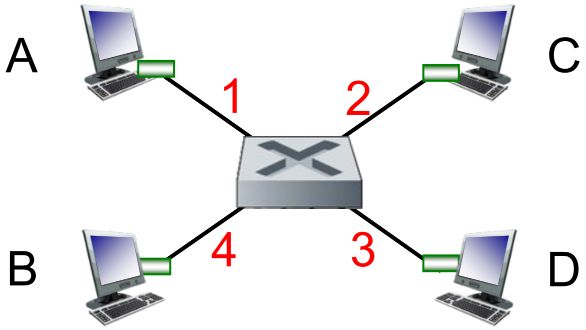


Switch Forwarding Table

- Selective Forwarding?
- Q: how does switch know A is reachable via interface 1?
  - A: each switch has a switch table.
    - Format of entry:

< MAC address of host, interface to reach host, TTL >

    - Looks like a routing table !

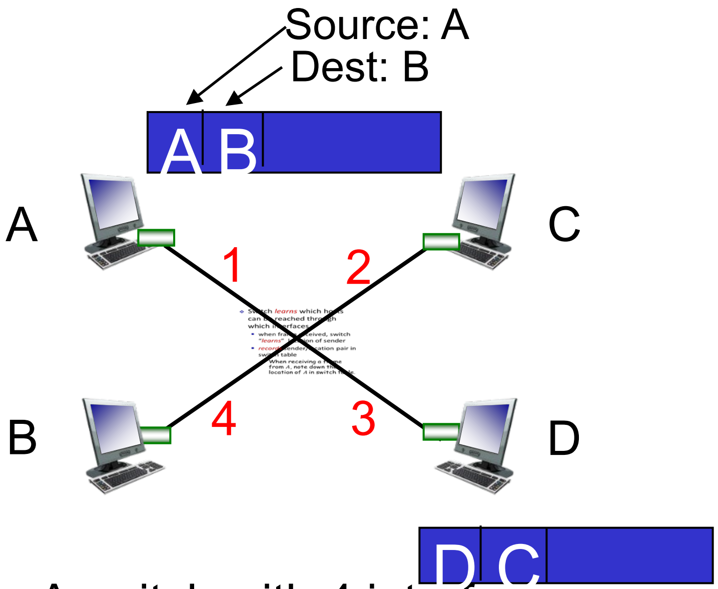


A switch with 4 interfaces (1, 2, 3, 4)

- Q: how are entries created and maintained in a switch table?
  - Something like a routing protocol?

Switch: Self-learning

- Switch *learns* which hosts can be reached through which interfaces.
  - when frame received, switch "*learns*" location of sender
  - *records* sender/location pair in switch table
    - When receiving a frame from A, note down the location of A in switch table.



A switch with 4 interfaces (1, 2, 3, 4)

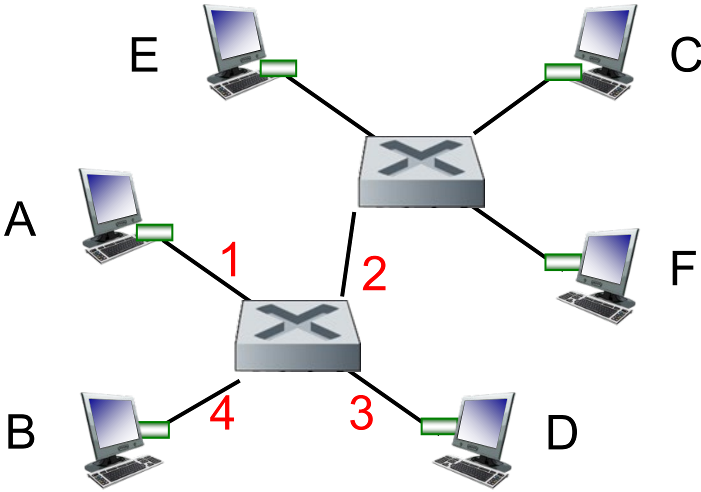
MAC addr	Interface	TTL
A	*1*	*60*
D	*3*	*60*

Switch table (initially empty)

Switch: frame filtering/forwarding

Let us try to *design* the forwarding algorithm

- Received a frame to *A* on interface *4*
  - Forward to interface *1*
- Received a frame to *D* on interface *1*
  - Forward to interface *2,3,4 (all except 1)*
- Received a frame to *F* on interface *2*
  - *Filter* the frame *(drop the frame)*



MAC addr	Interface	TTL
A	1	20
B	4	56
C	2	10
F	2	60

Switch table

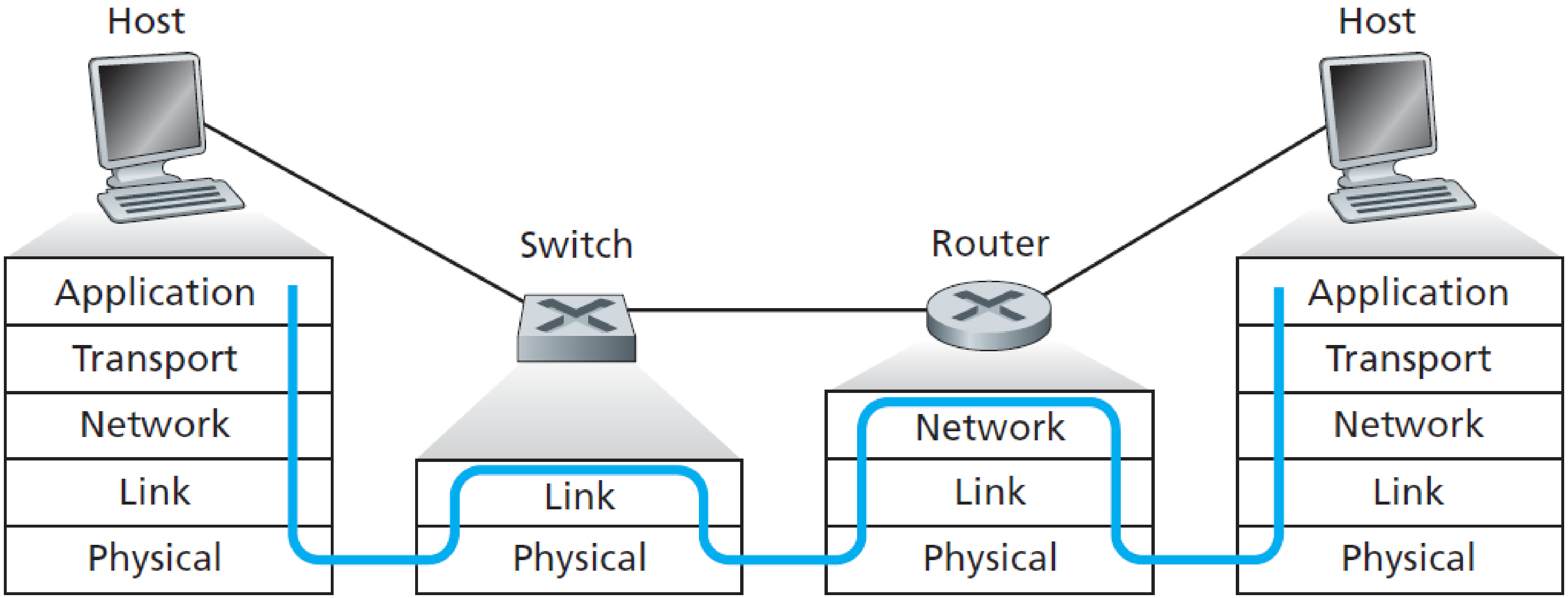
Switch: frame filtering/forwarding

When frame received at switch:

1. Record incoming link, MAC address of sending host
2. Index switch table using MAC destination address
3. *if* entry found for destination
    1. if destination on segment from which frame arrived
        1. drop frame
    2. else forward frame on interface indicated by entry
4. *else* flood
    1. forward on all interfaces except arriving interface

Switches vs. Routers

- Routers
  - Check IP address
  - Store-and-forward
  - Compute routes to destination
- Switches
  - Check MAC address
  - Store-and-forward
  - Forward frame to outgoing link or broadcast



Roadmap

6.1 Introduction to the Link Layer

6.2 Error Detection and Correction

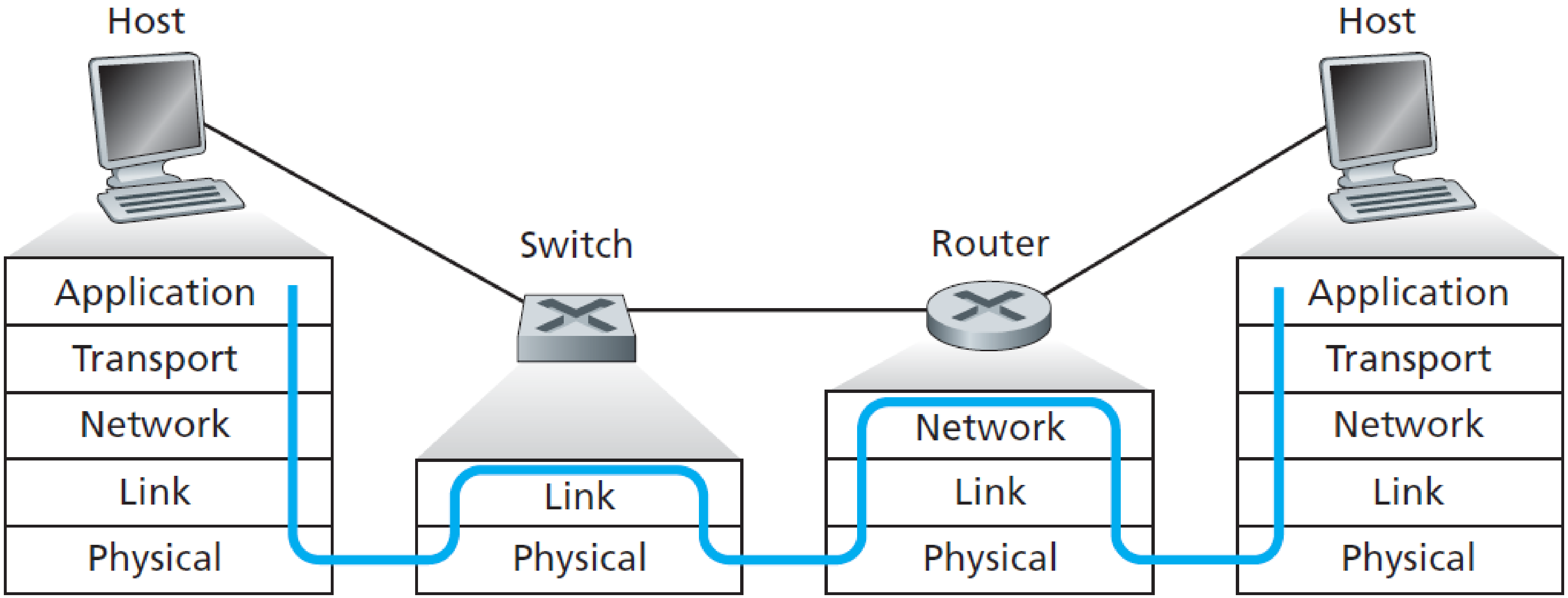
6.3 Multiple Access Links and Protocols

6.4 Switched Local Area Networks

- 6.4.1 Link Layer Addressing & ARP
- 6.4.2 Ethernet
- 6.4.3 Link-layer Switches

Switches vs. Routers

- Routers
  - Check IP address
  - Store-and-forward
  - Compute routes to destination
- Switches
  - Check MAC address
  - Store-and-forward
  - Forward frame to outgoing link or broadcast



Mac Address?

8 bytes	6	6	2	46 - 1500	4
Preamble	Dest Addr	Src Addr	Type	Data	CRC

- Question: How to know the MAC address of a receiving host, knowing its IP address?

ARP: Address Resolution Protocol

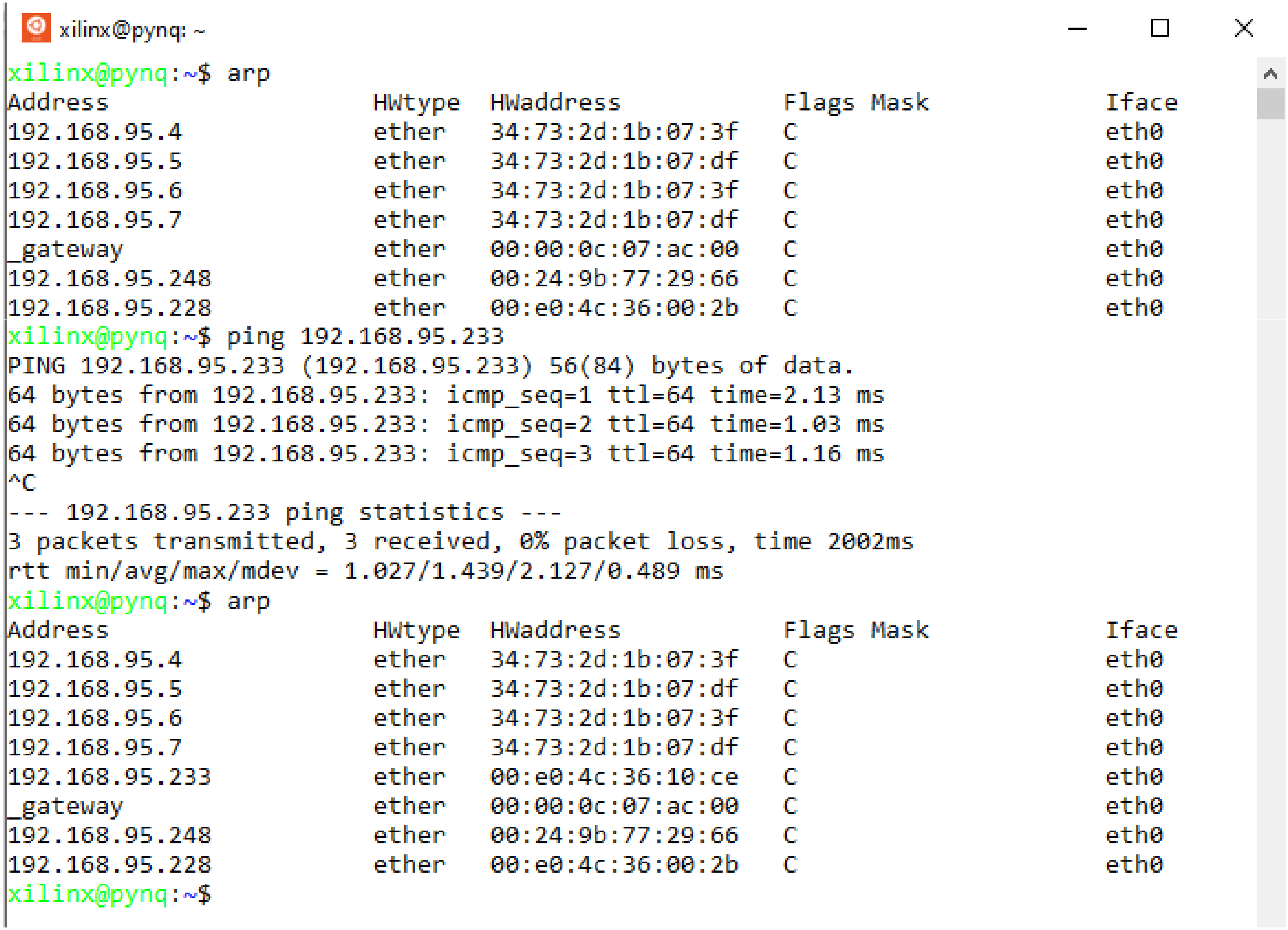
- Question: How to know the MAC address of a receiving host, knowing its IP address?
  - Use ARP [RFC 826]
    - Provides a *query* mechanism to learn the MAC address

- Each IP node has an ARP table.
  - Stores the mappings of IP address and MAC address of other nodes in the same subnet.

< IP address; MAC address; TTL >

time after which address mapping will be forgotten (typically a few minutes)

ARP Demo: Linux Device

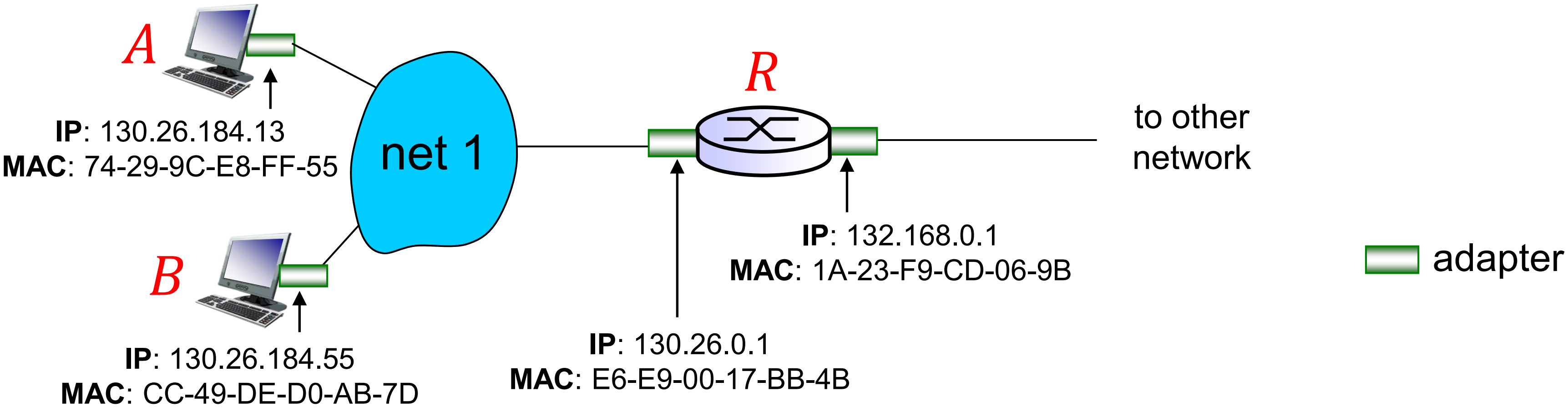


```
xilinx@pynq:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.95.4             ether   34:73:2d:1b:07:3f   C                     eth0
192.168.95.5             ether   34:73:2d:1b:07:df   C                     eth0
192.168.95.6             ether   34:73:2d:1b:07:3f   C                     eth0
192.168.95.7             ether   34:73:2d:1b:07:df   C                     eth0
_gateway                 ether   00:00:0c:07:ac:00   C                     eth0
192.168.95.248           ether   00:24:9b:77:29:66   C                     eth0
192.168.95.228           ether   00:e0:4c:36:00:2b   C                     eth0
xilinx@pynq:~$ ping 192.168.95.233
PING 192.168.95.233 (192.168.95.233) 56(84) bytes of data.
64 bytes from 192.168.95.233: icmp_seq=1 ttl=64 time=2.13 ms
64 bytes from 192.168.95.233: icmp_seq=2 ttl=64 time=1.03 ms
64 bytes from 192.168.95.233: icmp_seq=3 ttl=64 time=1.16 ms
^C
--- 192.168.95.233 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.027/1.439/2.127/0.489 ms
xilinx@pynq:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.95.4             ether   34:73:2d:1b:07:3f   C                     eth0
192.168.95.5             ether   34:73:2d:1b:07:df   C                     eth0
192.168.95.6             ether   34:73:2d:1b:07:3f   C                     eth0
192.168.95.7             ether   34:73:2d:1b:07:df   C                     eth0
192.168.95.233           ether   00:e0:4c:36:10:ce   C                     eth0
_gateway                 ether   00:00:0c:07:ac:00   C                     eth0
192.168.95.248           ether   00:24:9b:77:29:66   C                     eth0
192.168.95.228           ether   00:e0:4c:36:00:2b   C                     eth0
xilinx@pynq:~$
```

Sending Frame in the Same Subnet

- Suppose A wants to send data to B. They are in the same subnet.

  1. If A knows B's MAC address from its ARP table
     - create a frame with B's MAC addresses and send it.
     - Only B will process this frame.
     - Other nodes may receive but will ignore this frame.
  2. What if A is not aware of B's MAC address?

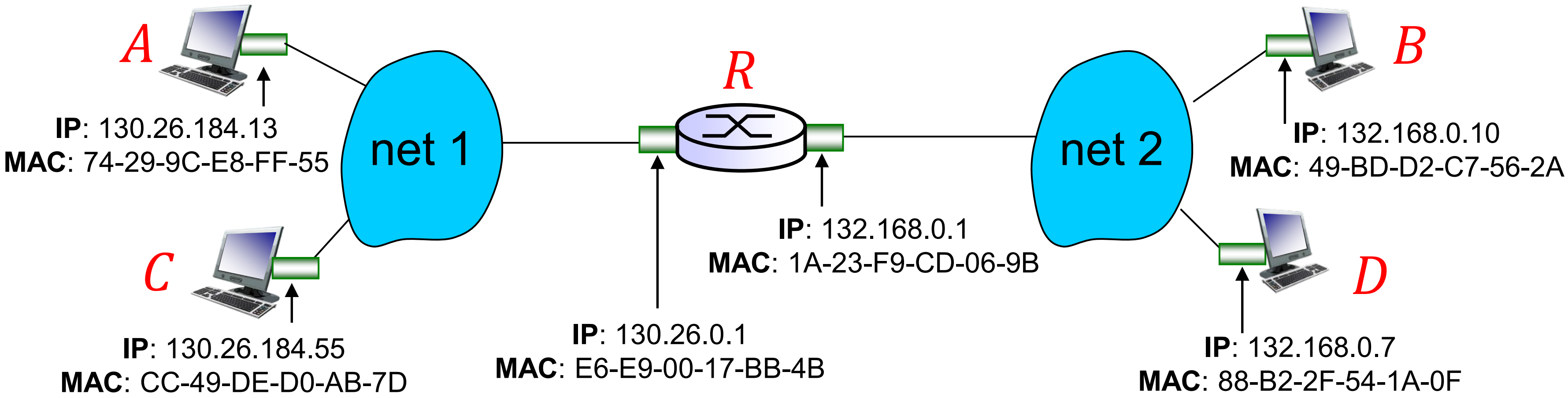


Sending Frame in the Same Subnet

- What if B's MAC address is not in A's ARP table?
  1. A broadcasts an ARP query packet, containing B's IP address.
     - Dest MAC address set to FF-FF-FF-FF-FF-FF
     - All the other nodes in the same subnet will receive this ARP query packet, but only B will reply to it.
  2. B *replies* to A with its MAC address.
     - Reply frame is sent to A's MAC address.
  3. A *caches* B's IP-to-MAC address mapping in its ARP table (until TTL expires).
- ARP is *"plug-and-play"*:
  - nodes create their ARP tables *without intervention from network administrator*

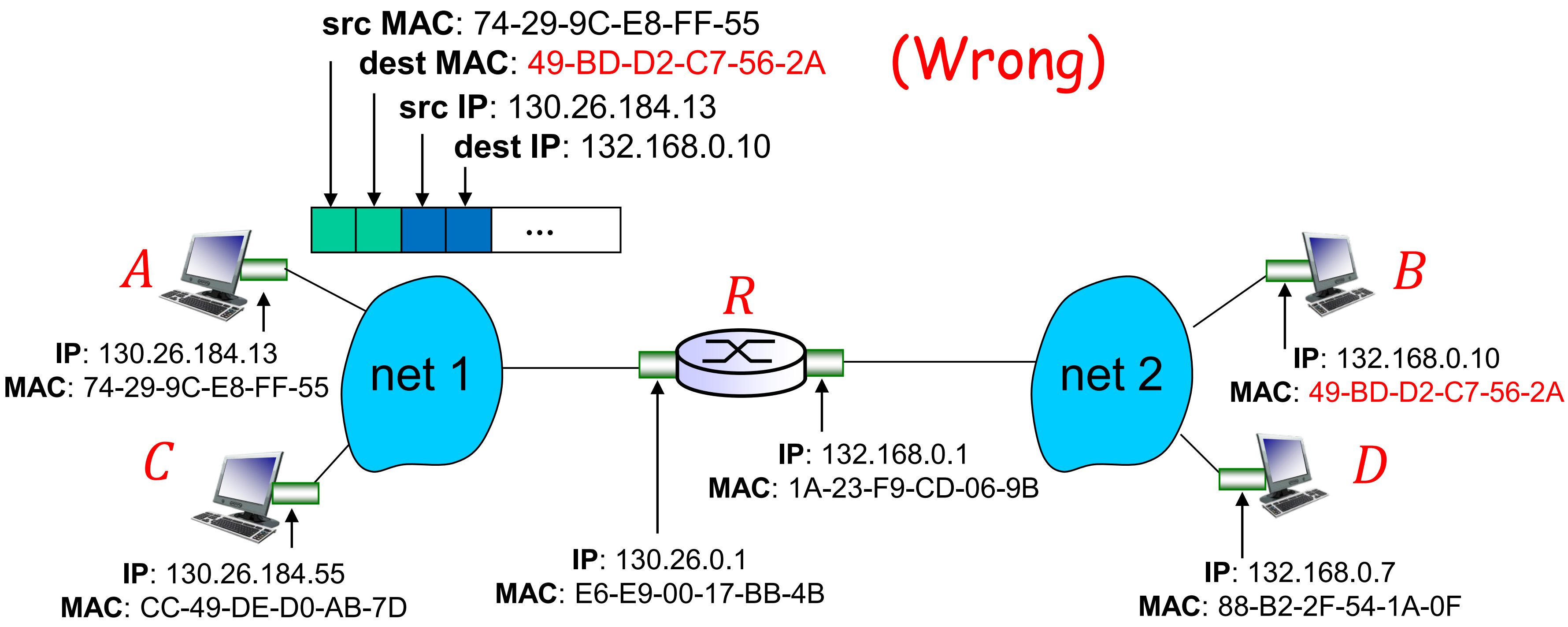
Sending Frame to Another Subnet

- Question: What if we send data to a host in another subnet?
  - For example, A sends datagram to B in another subnet.



Sending Frame to Another Subnet

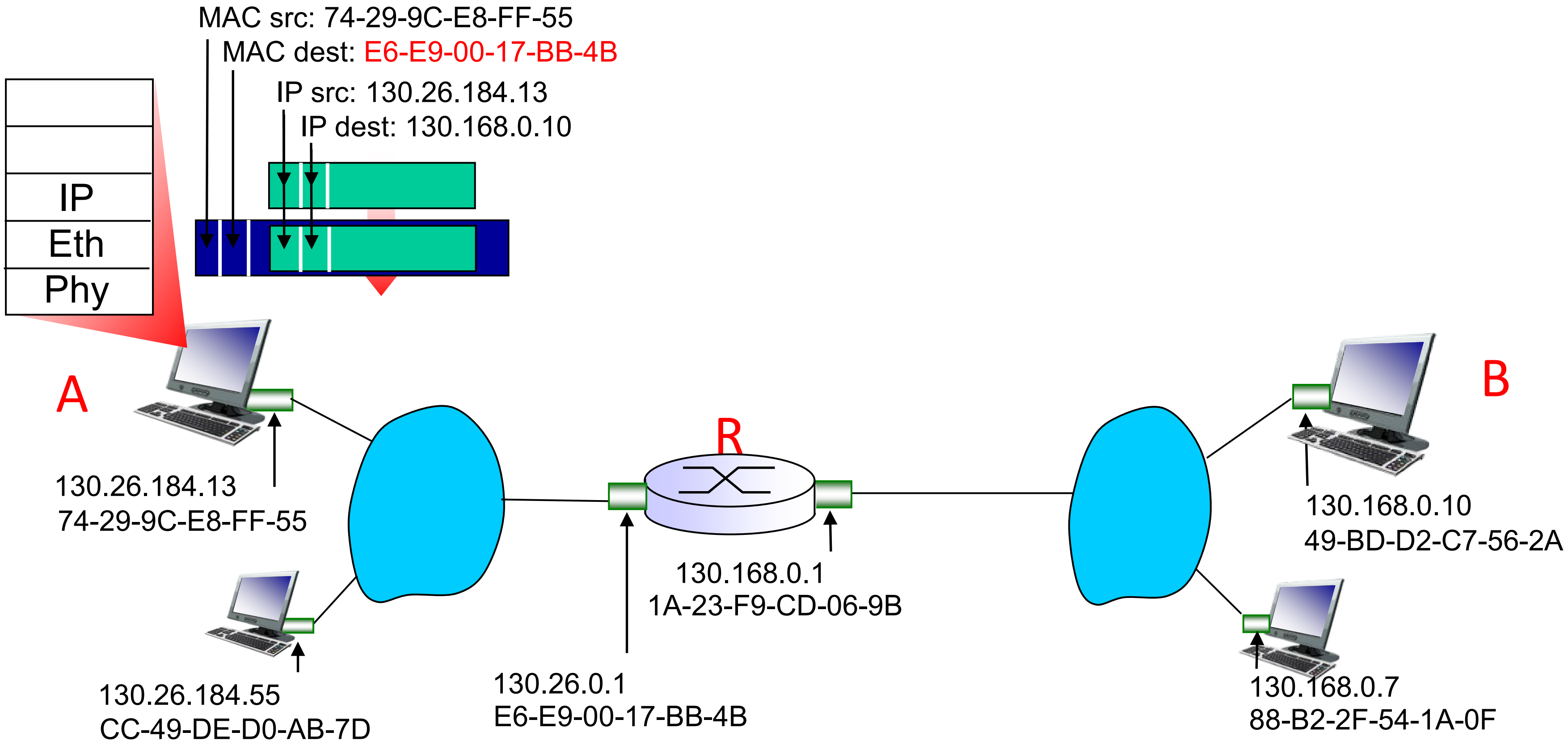
- *A* creates IP datagram with IP source *A*, destination *B*
- *A* creates a frame as follows
  - FALSE. all adapters in net 1 will ignore this frame because of the mismatch of destination MAC address.



(Wrong)

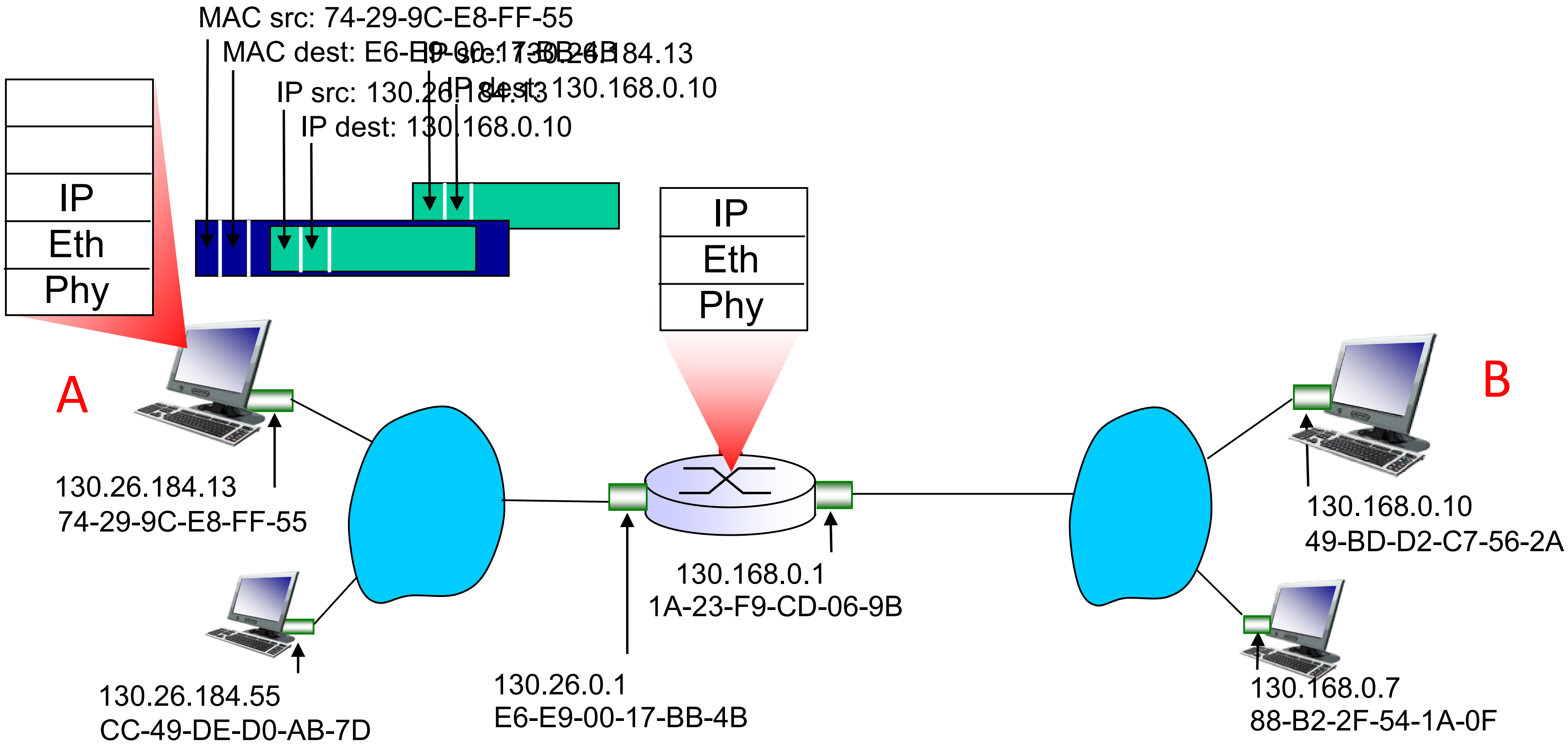
Sending Frame to Another Subnet

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



Sending Frame to Another Subnet

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



MAC src: 74-29-9C-E8-FF-55

MAC dest: E6-E9-00-17-BB-4B

IP src: 130.26.184.13

IP dest: 130.168.0.10

IP src: 130.26.184.13

IP dest: 130.168.0.10

IP

Eth

Phy

IP

Eth

Phy

A

B

130.26.184.13
74-29-9C-E8-FF-55

130.26.184.55
CC-49-DE-D0-AB-7D

130.26.0.1
E6-E9-00-17-BB-4B

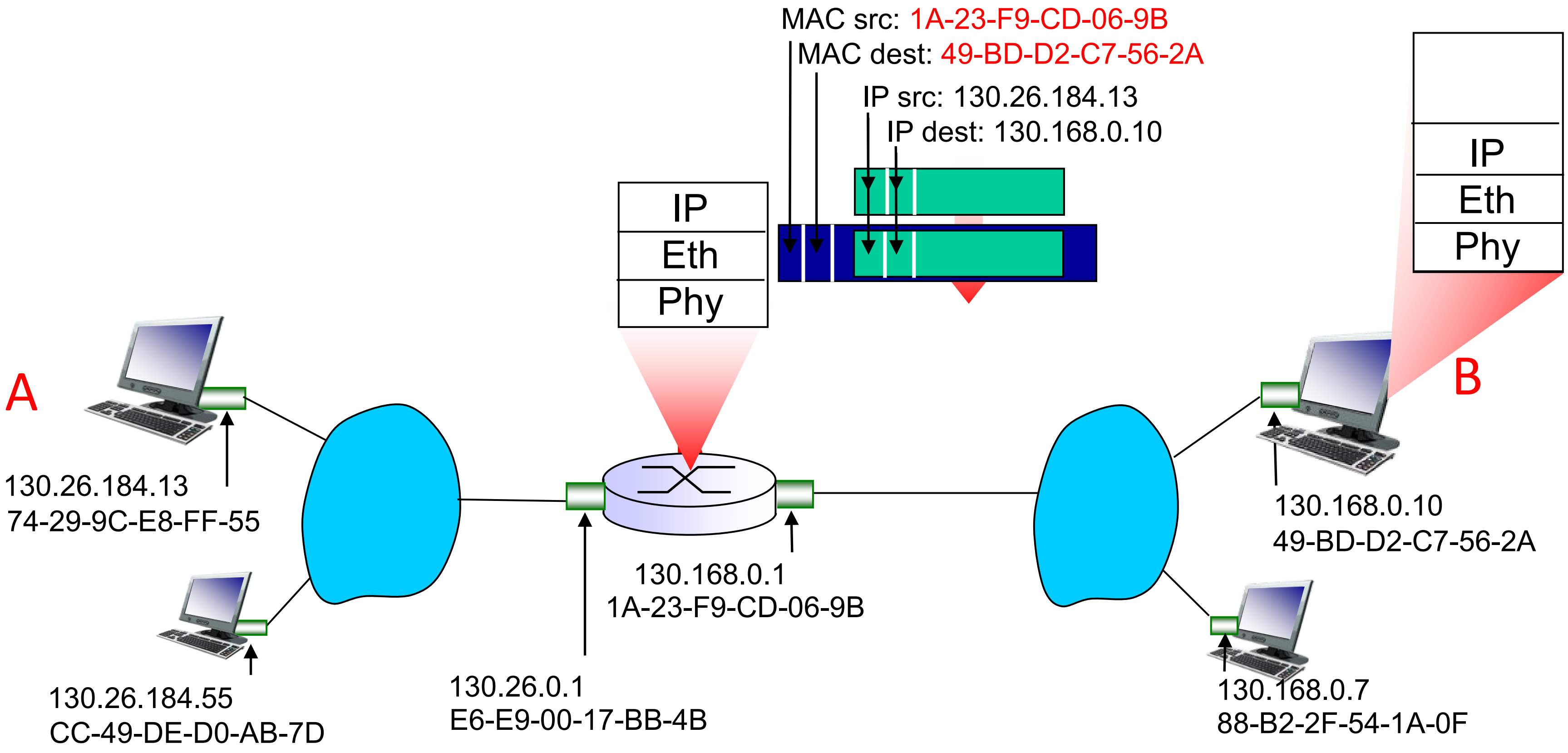
130.168.0.1
1A-23-F9-CD-06-9B

130.168.0.10
49-BD-D2-C7-56-2A

130.168.0.7
88-B2-2F-54-1A-0F

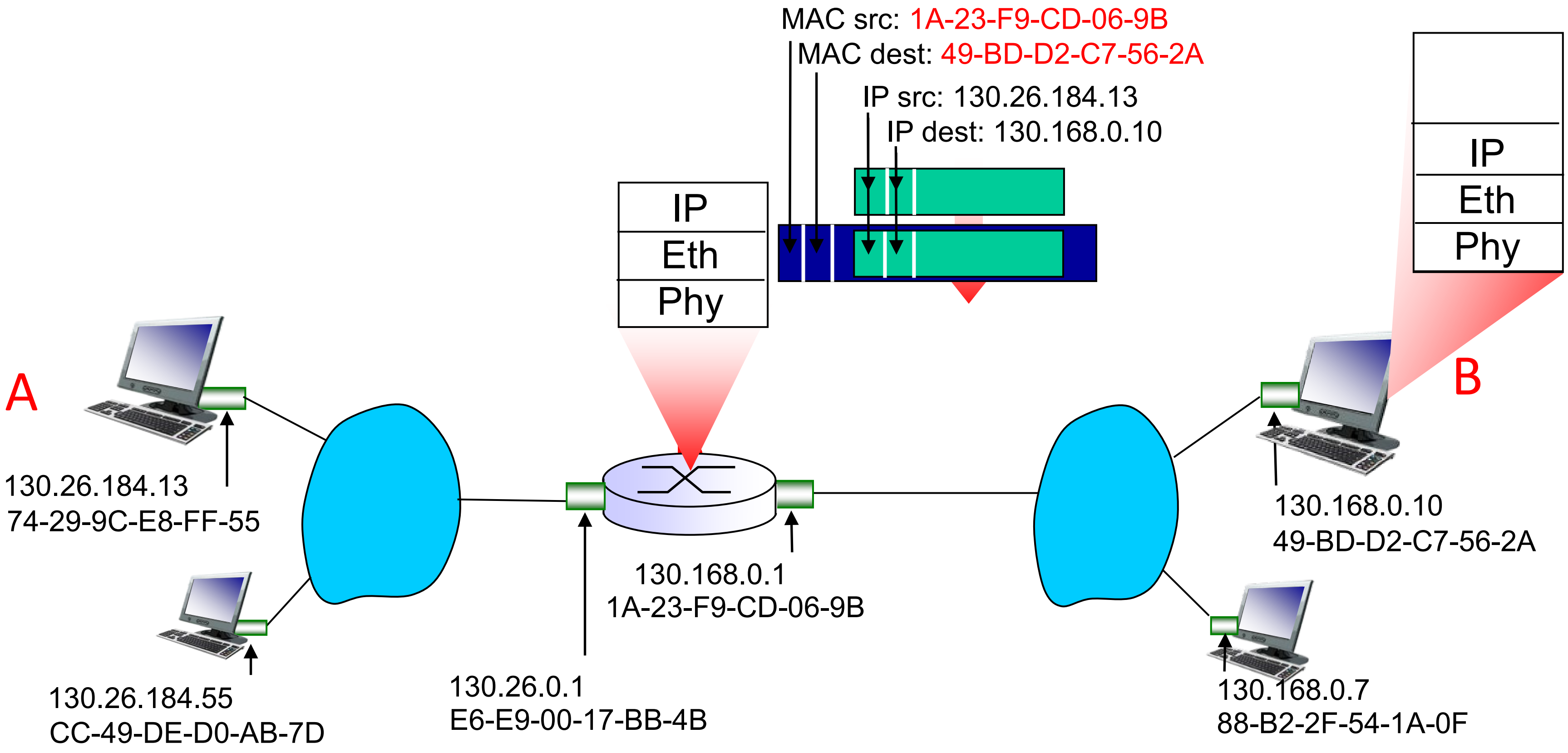
Sending Frame to Another Subnet

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



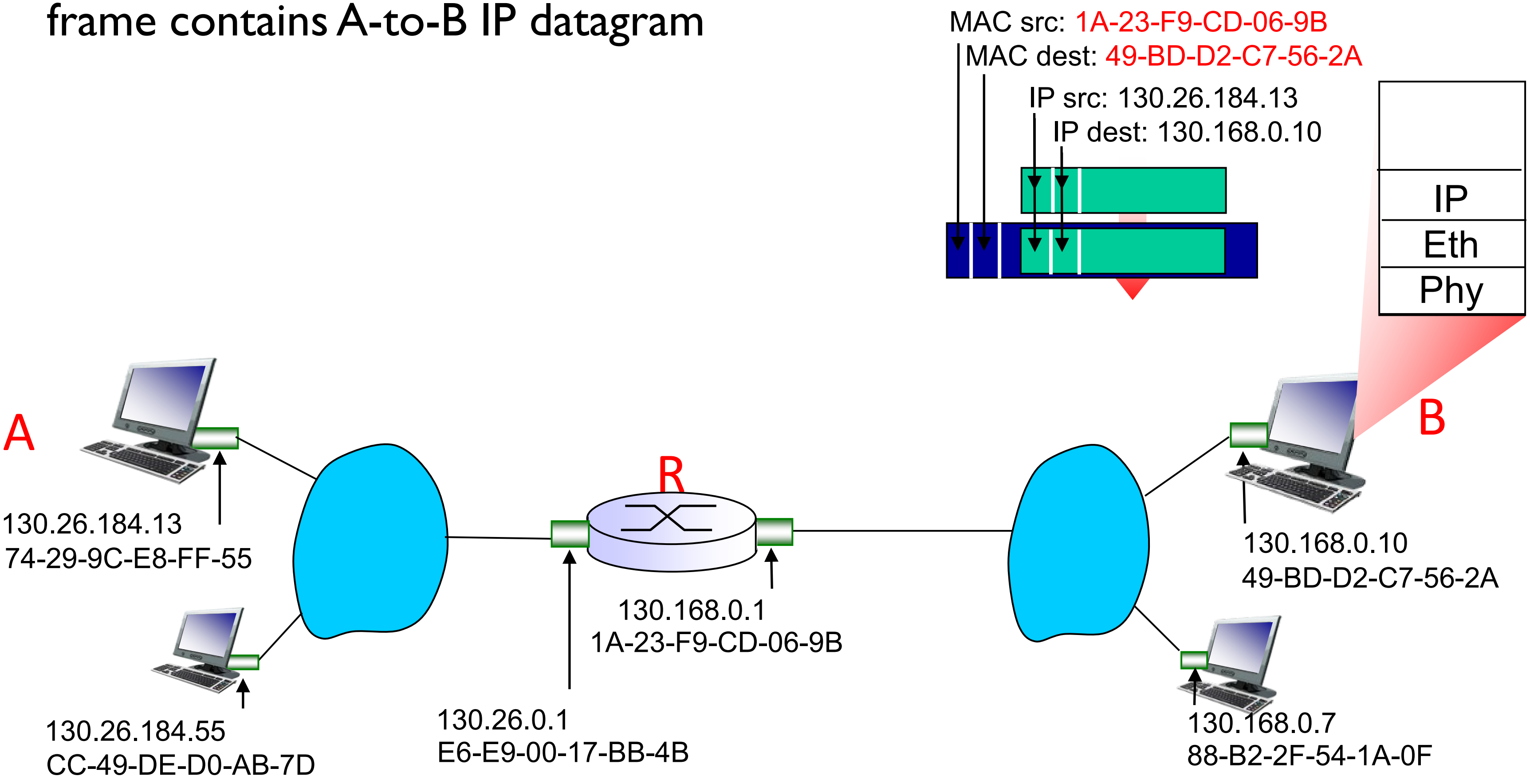
Sending Frame to Another Subnet

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



Sending Frame to Another Subnet

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



IP Address vs. MAC Address

- IP address
  - 32 bits in length
  - network-layer address used to move datagrams from source to dest.
  - Dynamically assigned; hierarchical (to facilitate routing)
  - Analogy: postal address

- MAC address
  - 48 bits in length
  - link-layer address used to move frames over every single link.
  - Permanent, to identify the hardware (adapter)
  - Analogy: NRIC number

Summary

- ARP [RFC 826] resolves the mapping from network layer (IP) address to link layer (MAC) address.

- Instantiation and implementation of link layer technologies.
  - Ethernet
  - Ethernet switches and switch tables

Chapter 6: let' s take a deep breath

- journey down protocol stack *complete* (except PHY)
- *solid* understanding of networking principles, practice
- ….. could stop here …. but *lots* of interesting topics!
  - wireless
  - multimedia
  - security