

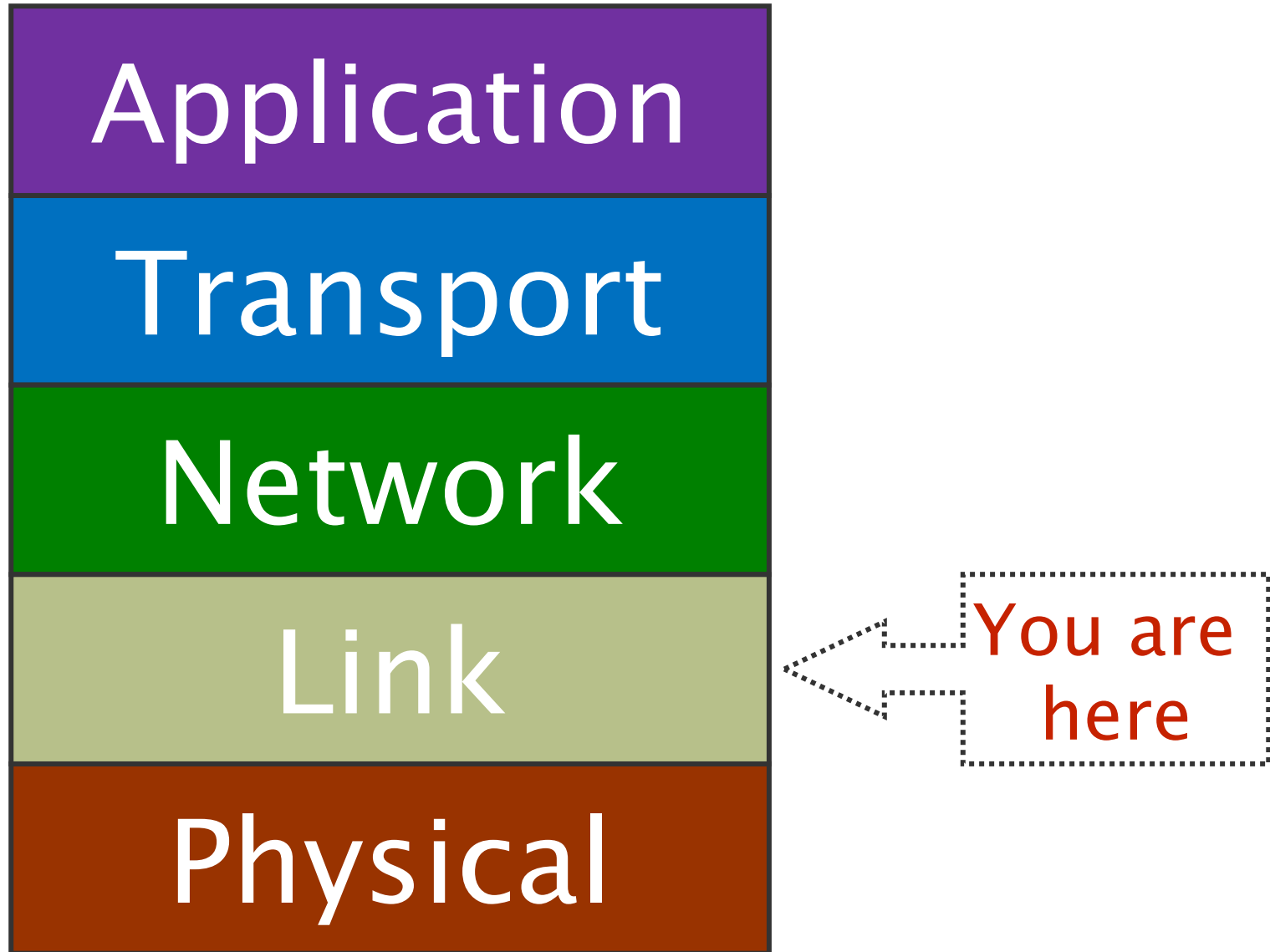
# CS2105

## An Awesome Introduction to Computer Networks

### Lecture 9: The Link Layer, Part II



Department of Computer Science  
School of Computing



# Lecture 9: The Link Layer

*After this class, you are expected to understand:*

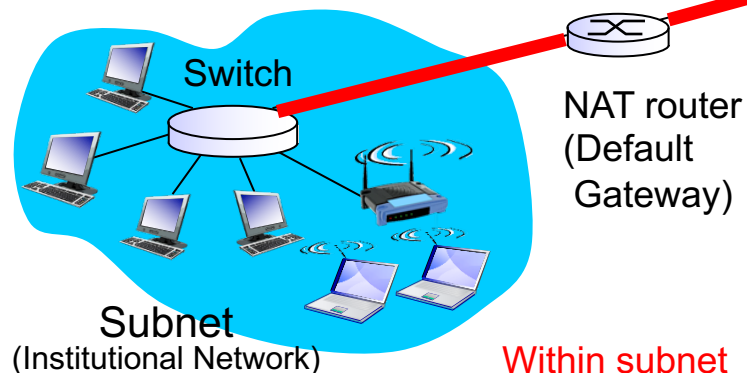
- ❖ the role of link layer and the services it could provide.
- ❖ how parity and CRC scheme work.
- ❖ different methods for accessing shared medium.
- ❖ how ARP allows a host to discover the MAC addresses of other nodes in the same subnet.
- ❖ the role of switches in interconnecting subnets in a LAN.

L8

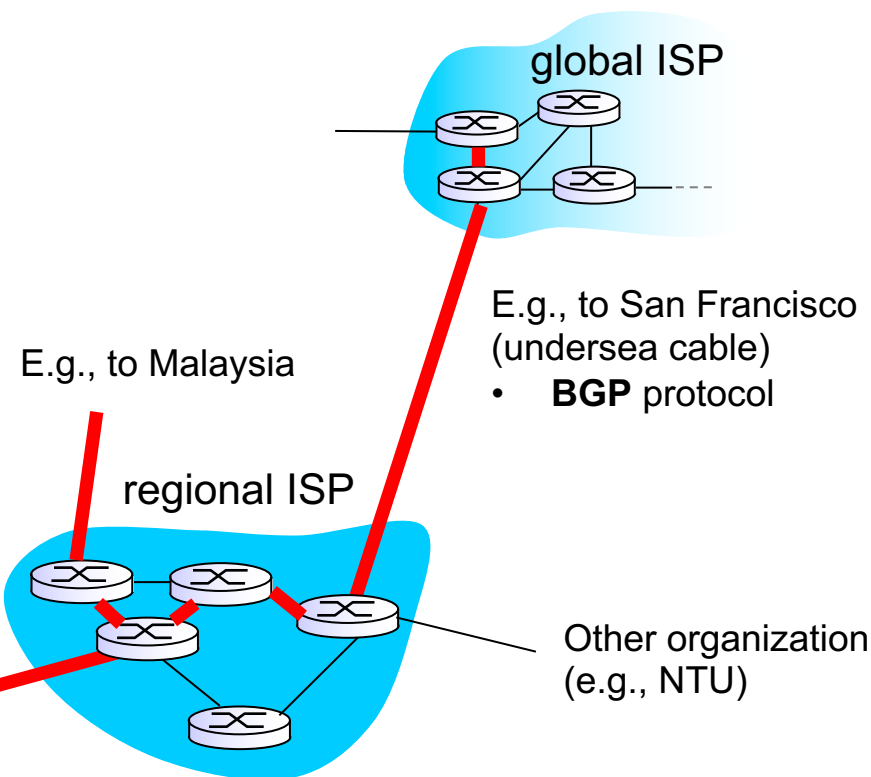
# Routing: Big Picture

**DHCP** protocol provides:

- IP address, e.g., 192.168.0.x/24
- Subnet mask, e.g., 255.255.255.0
- IP of DNS server
- IP of Default Gateway (e.g.: 192.168.0.1)



Within subnet  
• **ARP** protocol



Which link/path to choose?

Intra-AS routing

- **RIP, OSPF** protocols
- Distributed algo.
- Build routing table

# Lecture 8&9: Roadmap

6.1 Introduction to the Link Layer

6.2 Error Detection and Correction

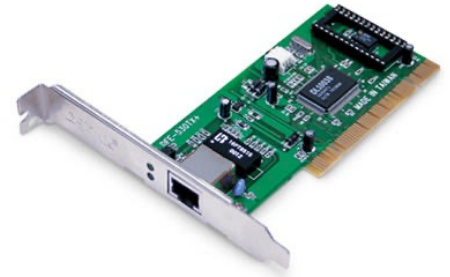
6.3 Multiple Access Links and Protocols

6.4 Switched Local Area Networks

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

# MAC Address (1/2)

- ❖ 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 (2/2)

- ❖ MAC address is typically 48 bits, burned in NIC ROM (sometimes software settable).
  - Example: **5C-F9-DD-E8-E3-D2** — hexadecimal (base 16) notation
  - MAC address allocation is administered by IEEE.
    - The first three bytes identifies the vendor of an adapter.
  - Several websites allow us to check the vendor given a MAC address, e.g.:

<https://macvendors.com/>

# MAC Address (Ex: 1/3)

macvendors.com

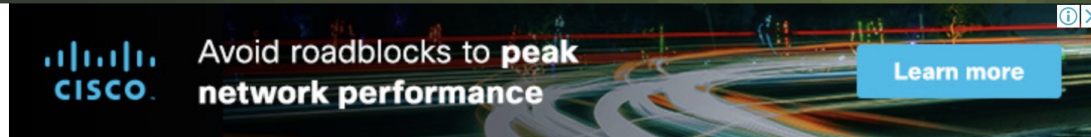


MA:CV:en:do:rs

[Home](#) [API](#) [Plans](#) [About](#) [Register](#) [Login](#)

## Find MAC Address Vendors. Now.

Enter a MAC Address



### // Features



#### Data

Our list of vendors is provided directly from the IEEE Standards Association and is updated multiple times each day. The IEEE is the registration authority and provides us data on over 16,500 registered vendors.



#### Speed

Our API was designed from the ground up with performance in mind. We have stripped our API down to the bare essentials, optimized our servers, and organized our data so that whether your app is making 100 requests a day, or 100,000, you'll never be left waiting.



#### Simple

We have eliminated all unnecessary overhead from our systems. Simply send us an HTTP GET/POST request with your MAC address and we'll return the



#### Reliable

We want you to feel comfortable building your systems around ours. Since launching in 2011, we have grown at an incredible pace. Today our API receives



# MAC Address (Ex: 2/3)

macvendors.com

MA:CV:en:do:rs

[Home](#) [API](#) [Plans](#) [About](#) [Register](#) [Login](#)

## Find MAC Address Vendors. Now.

Enter a MAC Address

5C:F9:DD:E8:E3:D2

Dell Inc.



Avoid roadblocks to **peak**  
network performance

[Learn more](#)

### // Features



#### Data

Our list of vendors is provided directly from the IEEE Standards Association and is updated multiple times each day. The IEEE is the registration authority and provides us data on over 16,500 registered vendors.



#### Speed

Our API was designed from the ground up with performance in mind. We have stripped our API down to the bare essentials, optimized our servers, and organized our data so that whether your app is making 100 requests a day, or 100,000, you'll never be left waiting.



#### Simple

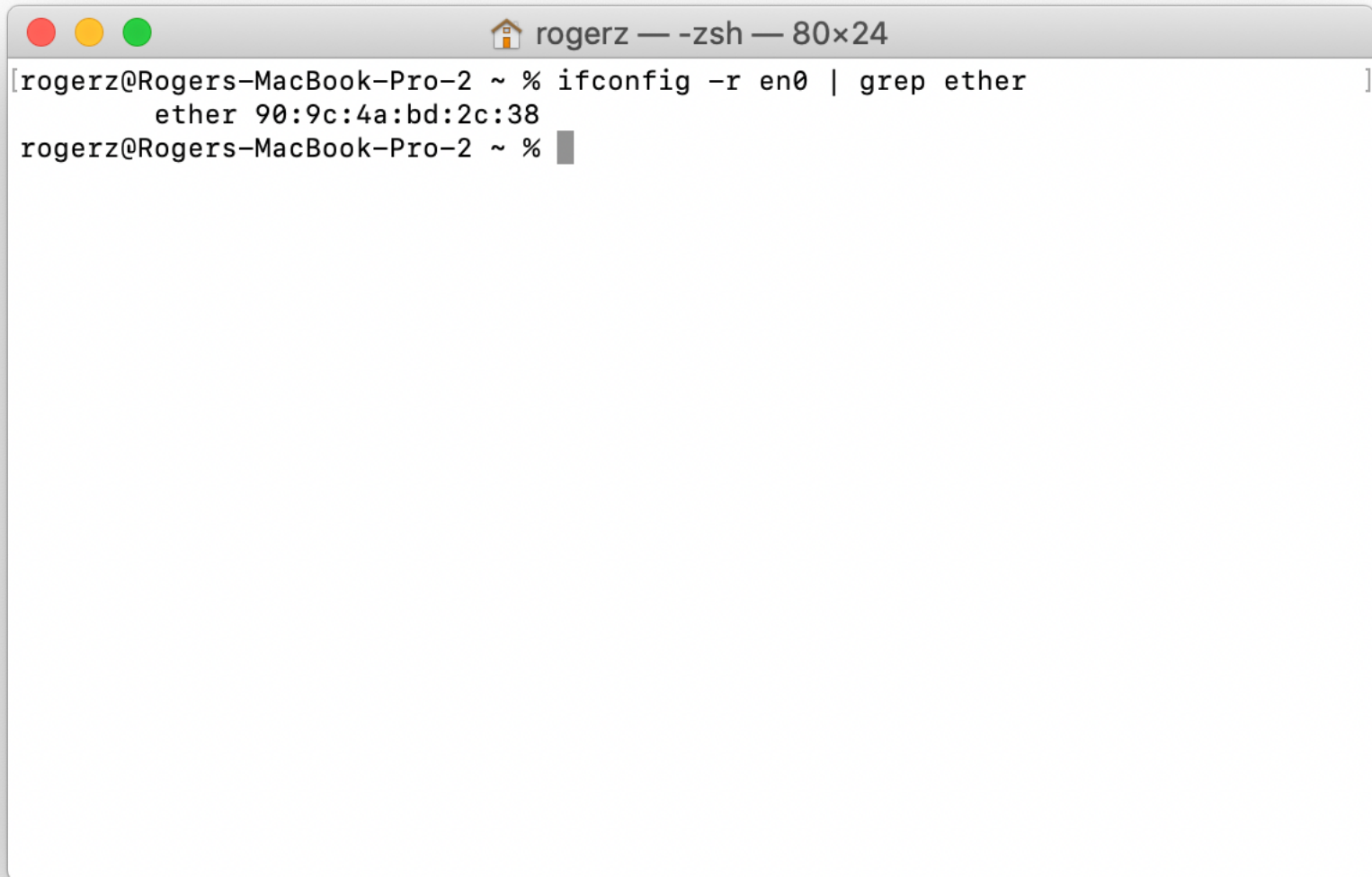
We have eliminated all unnecessary overhead from our systems. Simply send us an HTTP GET/POST request with your MAC address and we'll return the



#### Reliable

We want you to feel comfortable building your systems around ours. Since launching in 2011, we have grown at an incredible pace. Today our API receives

# MAC Address (Ex: 3/3)

A terminal window titled "rogerz — -zsh — 80x24" with a home icon. The window shows a command being executed: "ifconfig -r en0 | grep ether". The output is "ether 90:9c:4a:bd:2c:38". The prompt "rogerz@Rogers-MacBook-Pro-2 ~ %" is visible at the end of the line.

```
rogerz@Rogers-MacBook-Pro-2 ~ % ifconfig -r en0 | grep ether
ether 90:9c:4a:bd:2c:38
rogerz@Rogers-MacBook-Pro-2 ~ %
```

# 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**

# ARP: Address Resolution Protocol

- ❖ **Question:** How to know the MAC address of a receiving host, knowing its IP address?
  - Use ARP [RFC 826]
- ❖ Each IP node (host, router) has an **ARP table**.
  - Stores the mappings of IP address and MAC address of other nodes in the same subnet.

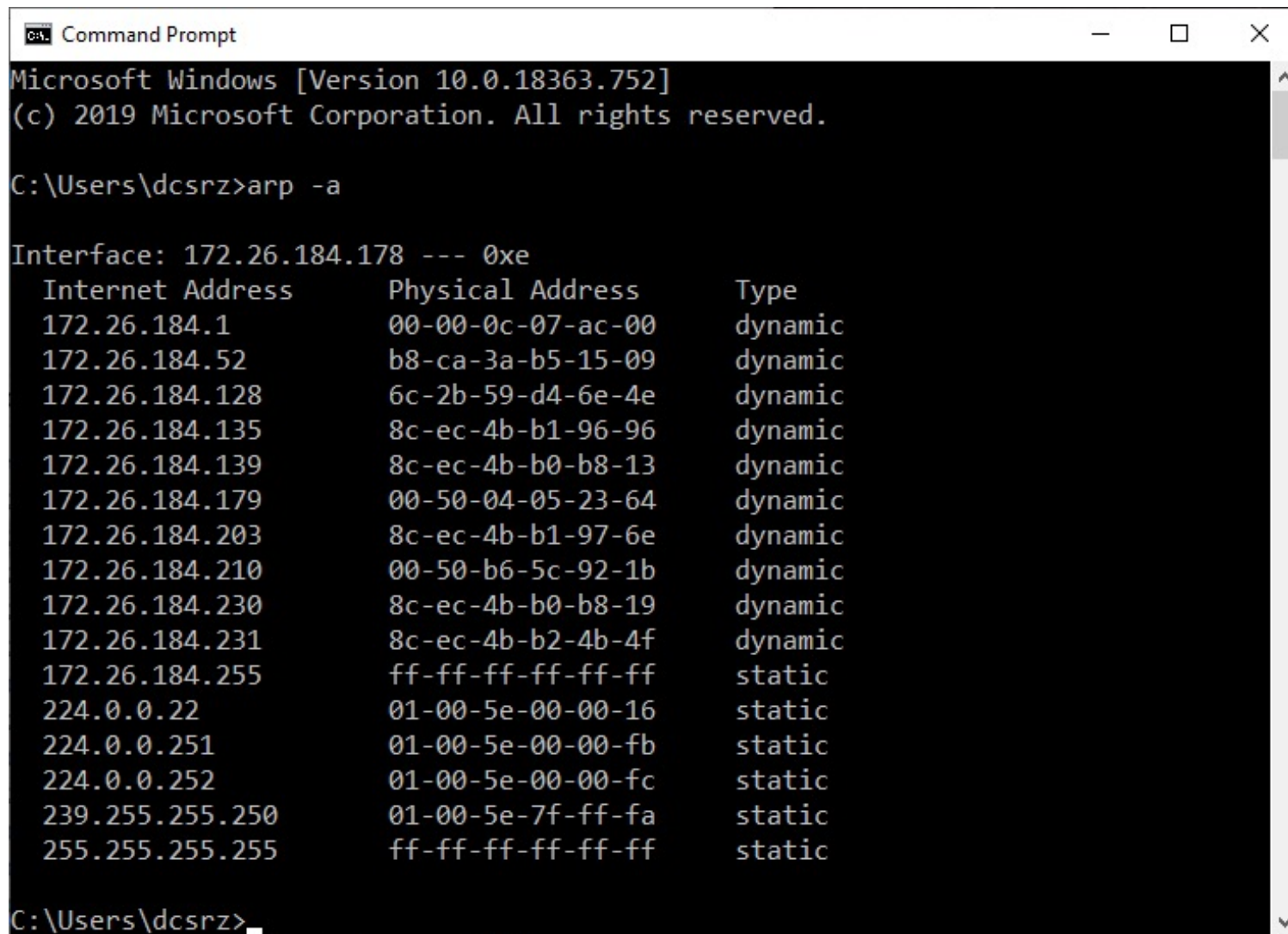
< IP address; MAC address; TTL >

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

*As Maverick (Top Gun) likes to say:*



# ARP Demo: Office Windows PC



```
Command Prompt
Microsoft Windows [Version 10.0.18363.752]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\dcsrcz>arp -a

Interface: 172.26.184.178 --- 0xe
    Internet Address      Physical Address      Type
    172.26.184.1          00-00-0c-07-ac-00     dynamic
    172.26.184.52         b8-ca-3a-b5-15-09     dynamic
    172.26.184.128        6c-2b-59-d4-6e-4e     dynamic
    172.26.184.135        8c-ec-4b-b1-96-96     dynamic
    172.26.184.139        8c-ec-4b-b0-b8-13     dynamic
    172.26.184.179        00-50-04-05-23-64     dynamic
    172.26.184.203        8c-ec-4b-b1-97-6e     dynamic
    172.26.184.210        00-50-b6-5c-92-1b     dynamic
    172.26.184.230        8c-ec-4b-b0-b8-19     dynamic
    172.26.184.231        8c-ec-4b-b2-4b-4f     dynamic
    172.26.184.255        ff-ff-ff-ff-ff-ff     static
    224.0.0.22            01-00-5e-00-00-16     static
    224.0.0.251           01-00-5e-00-00-fb     static
    224.0.0.252           01-00-5e-00-00-fc     static
    239.255.255.250       01-00-5e-7f-ff-fa     static
    255.255.255.255       ff-ff-ff-ff-ff-ff     static

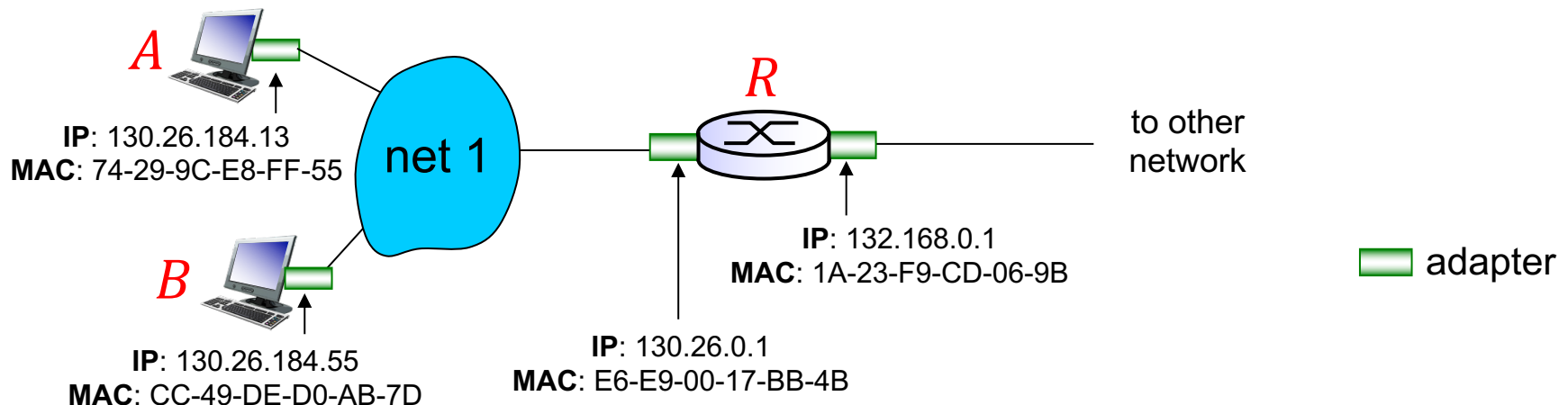
C:\Users\dcsrcz>
```

# 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?

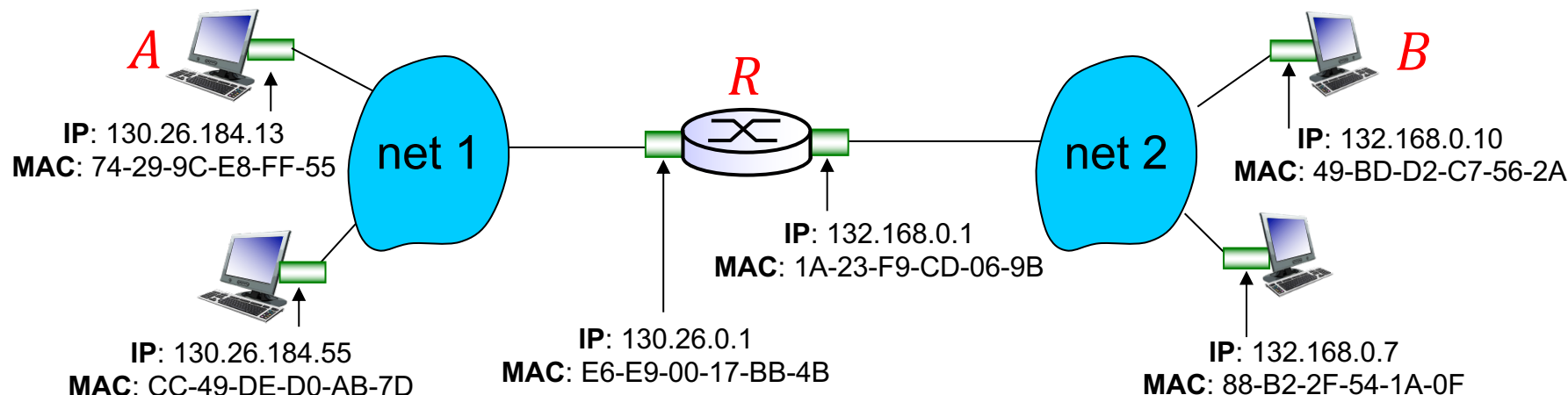
- ①  $A$  **broadcasts** an ARP query packet, containing  $B$ 's IP address.
  - Dest MAC address set to FF-FF-FF-FF-FF-FF
  - All the other nodes in the same subnet will receive this ARP query packet, but only  $B$  will reply it.
- ②  $B$  replies to  $A$  with its MAC address.
  - Reply frame is sent to  $A$ 's MAC address.
- ③  $A$  caches  $B$ 's IP-to-MAC address mapping in its ARP table (until TTL expires).

Question: how to determine if  $B$  is in the same subnet?



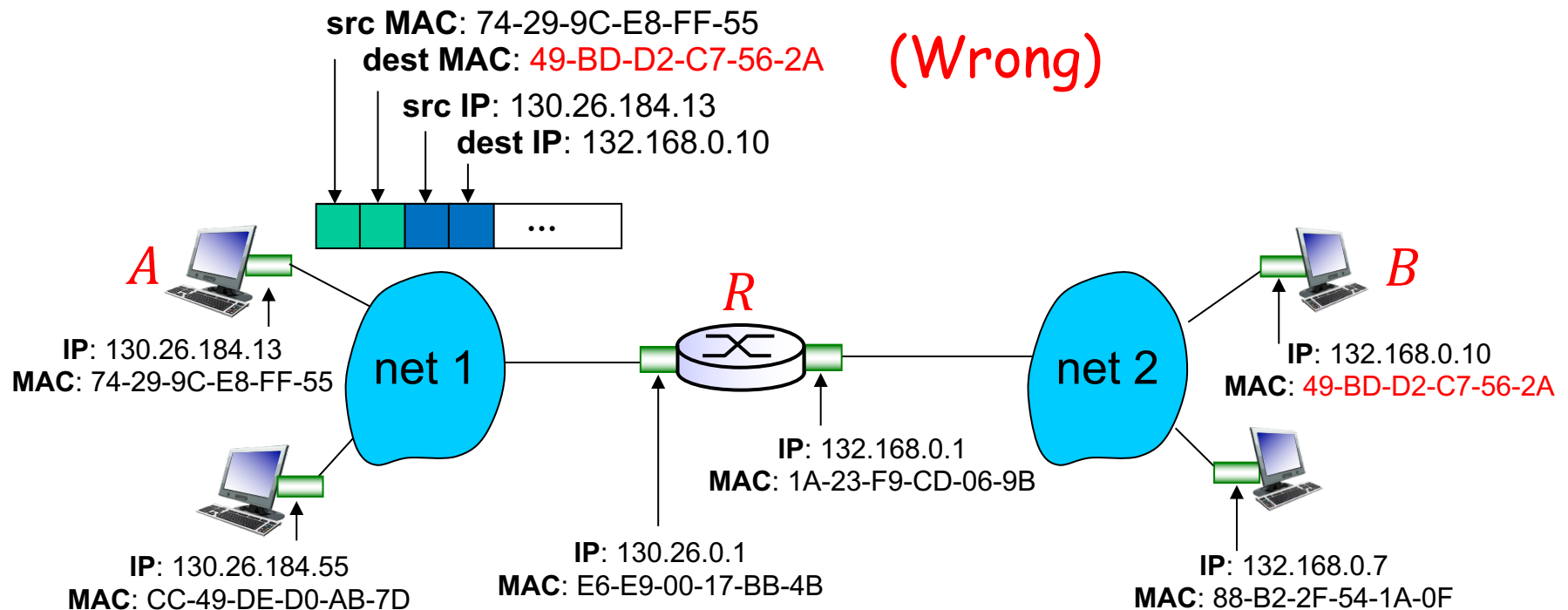
# Sending Frame to **Another** Subnet

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



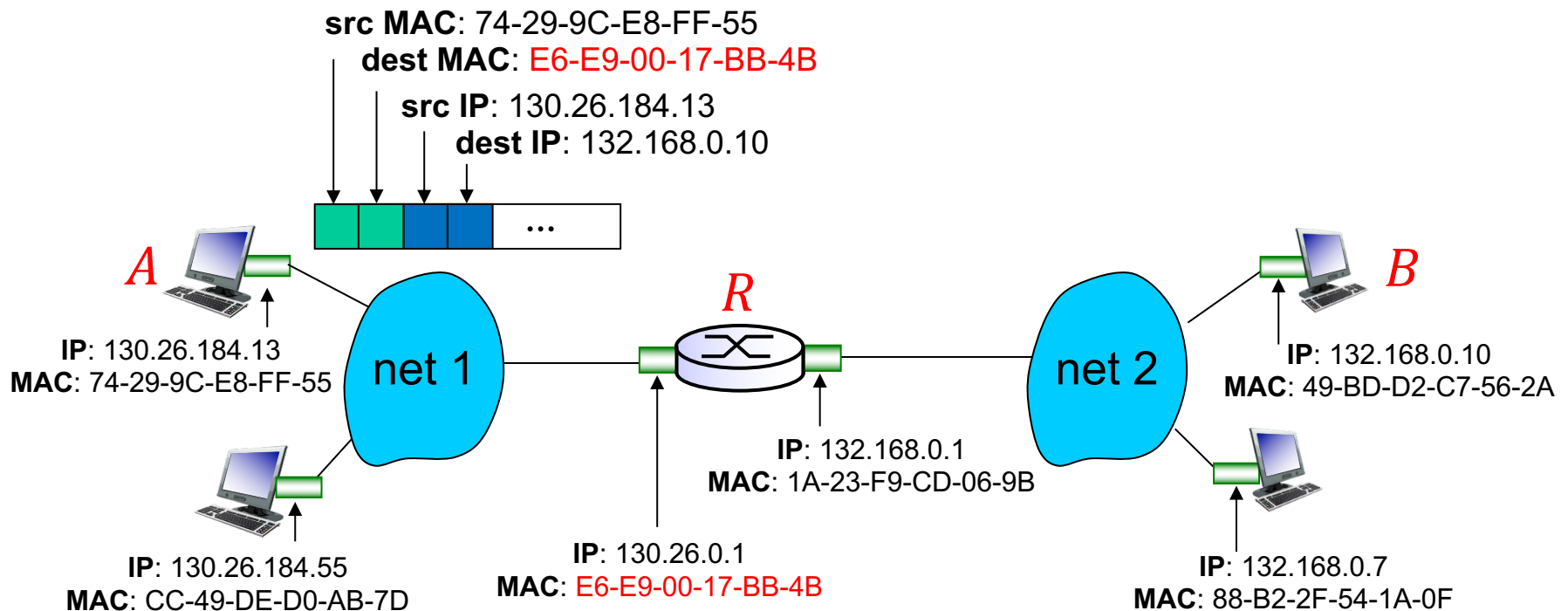
# Sending Frame to **Another** Subnet

- ❖ *A* sends datagram to *B* in another subnet.
  - Can *A* create a frame as follows?
    - **No.** all adapters in net 1 will ignore this frame because of the mismatch of destination MAC address.



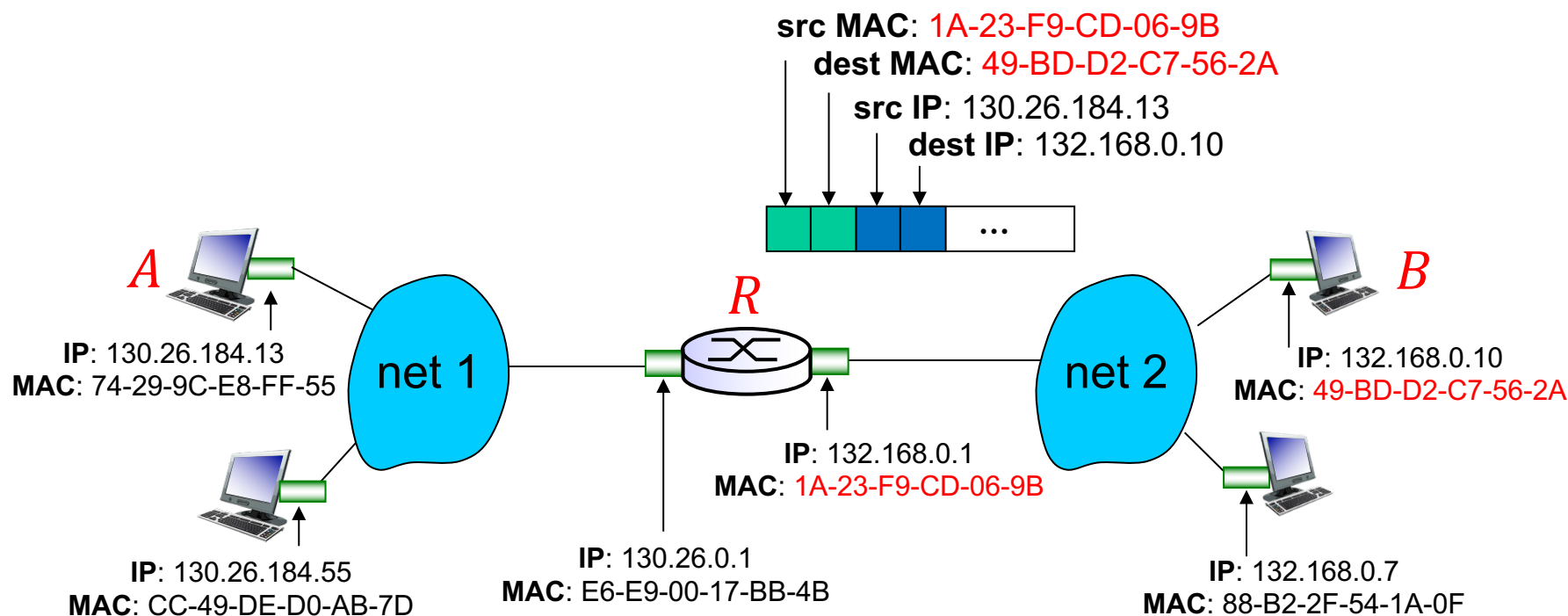
# Sending Frame to **Another** Subnet

- ❖ *A* sends datagram to *B* in another subnet.
  - *A* should create a link-layer frame with (1) *R*'s MAC address (2) *B*'s IP address as destination.



# Sending Frame to **Another** Subnet

- ❖ *A* sends datagram to *B* in another subnet.
  - *R* will move datagram to outgoing link and construct a new frame with *B*'s MAC address.



# Lectures 8&9: Roadmap

6.1 Introduction to the Link Layer

6.2 Error Detection and Correction

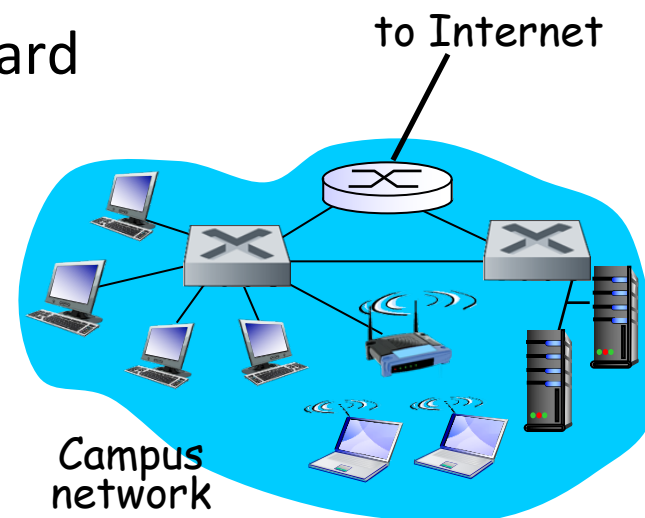
6.3 Multiple Access Links and Protocols

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

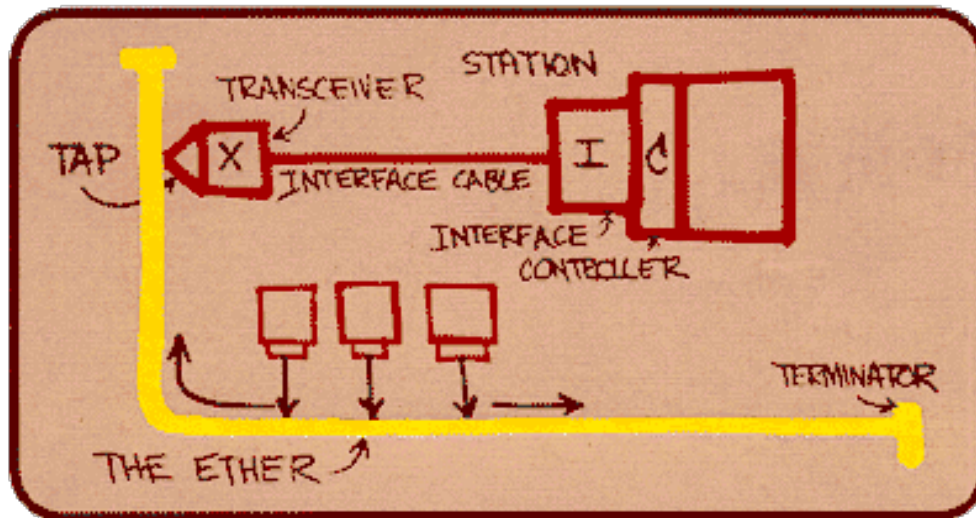


# Ethernet

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



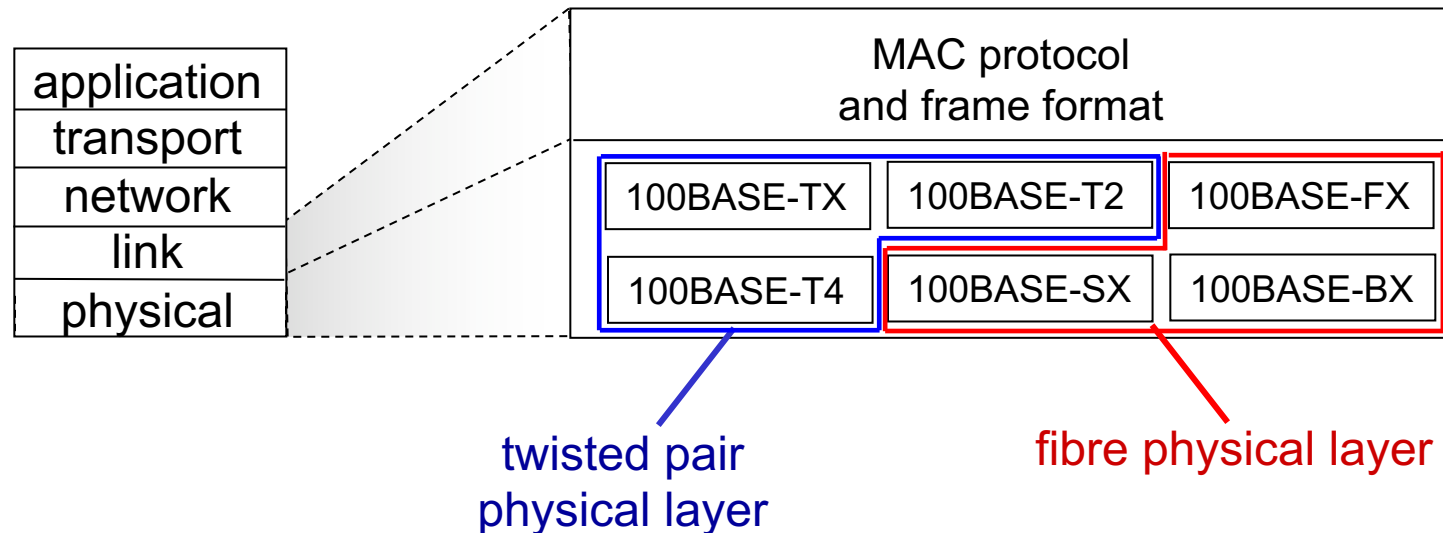
Ethernet connection  
(Source: Wikipedia)



*Metcalfe's  
Ethernet sketch*

# 802.3 Ethernet Standards (1/2)

- ❖ 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

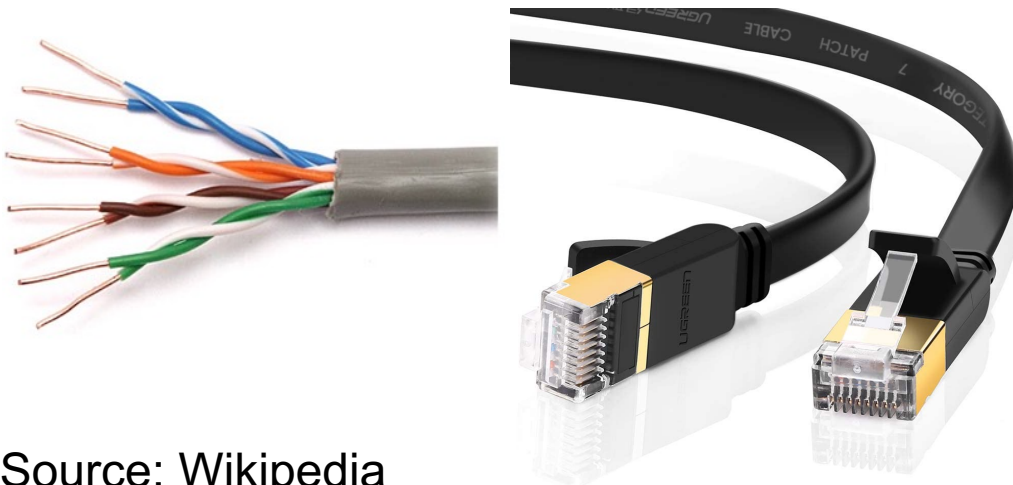




# 802.3 Ethernet Standards (2/2)

## ❖ Twisted Pair Copper Connectors:

- RJ45
- CAT 6
  - Max. speed: 10 Gbps
  - Max. length: 100m



Source: Wikipedia

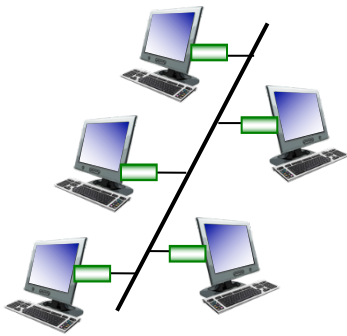
## ❖ Optical Fibre Connectors:

- Left: LC/PC connectors  
Right: SC/PC connectors
- Single-mode fibre
  - Max. speed: 10 or 40 Gbps
  - Max. length: > 80 km

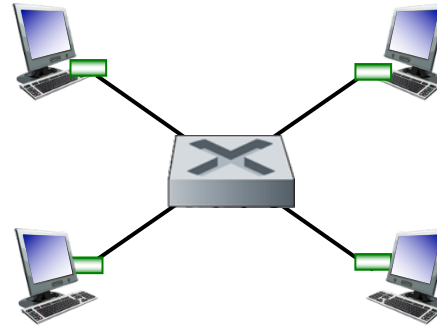


# Ethernet: Physical Topology

- ❖ **Bus** topology: popular in mid 90s
  - all nodes can collide with each other
- ❖ **Star** topology: prevails today
  - switch in center
  - nodes do not collide with each other



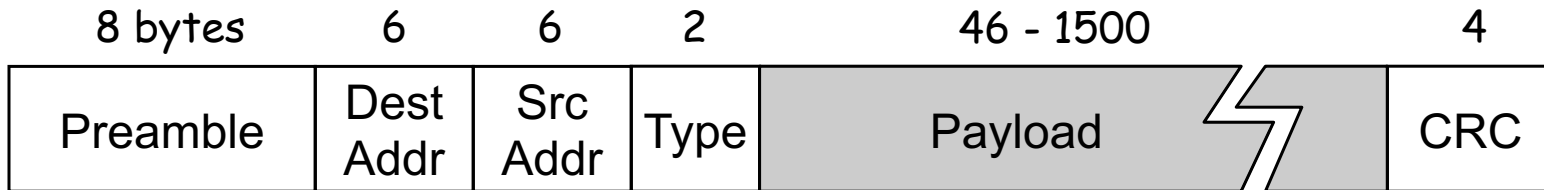
Ethernet with **bus** topology



Ethernet with **star** topology

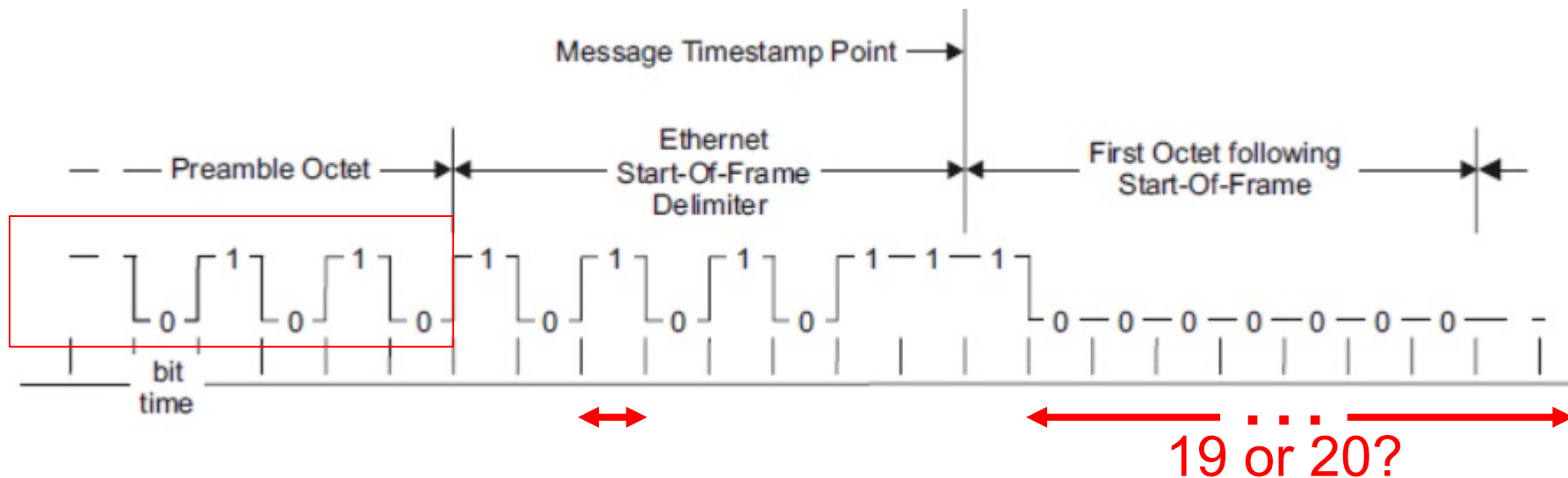
# Ethernet Frame Structure (1/3)

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



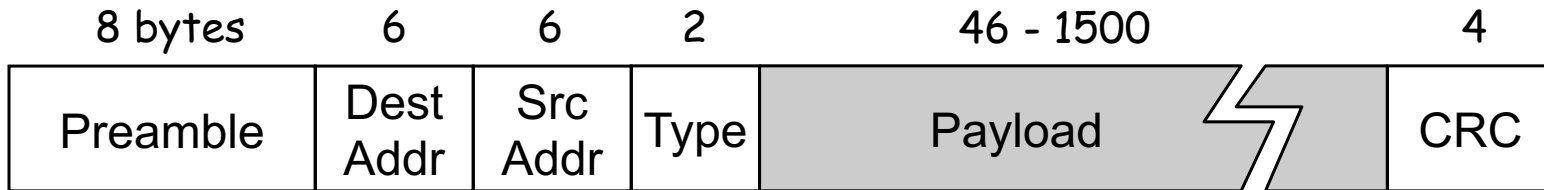
- ❖ *Preamble:*
  - 7 bytes with pattern **10101010** ( $AA_{Hex}$ ) followed by 1 byte with pattern **10101011** ( $AB_{Hex}$ ).
  - used to synchronize receiver and sender clock rates.

# Ethernet Frame Structure (2/3)



- ❖ The preamble provides a “square wave” pattern that tells the receiver the sender’s clock rate;
- ❖ and 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.

# Ethernet Frame Structure (3/3)



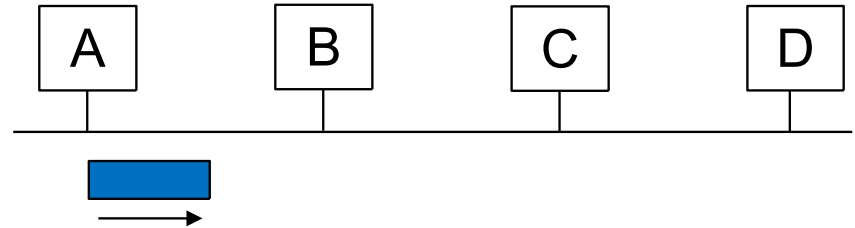
- ❖ *Source and dest MAC address:*
  - If NIC receives a frame with matching destination address, or with broadcast address, it passes data in the frame to network layer protocol.
  - Otherwise, NIC discards frame.
- ❖ *Type:* Indicates higher layer protocol (mostly IP).
- ❖ *CRC:* corrupted frame will be dropped.

# Ethernet Data Delivery Service

- ❖ **Connectionless**: no handshaking between sending and receiving NICs.
- ❖ **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.

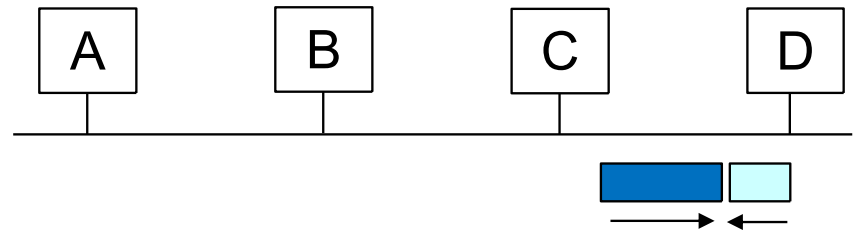
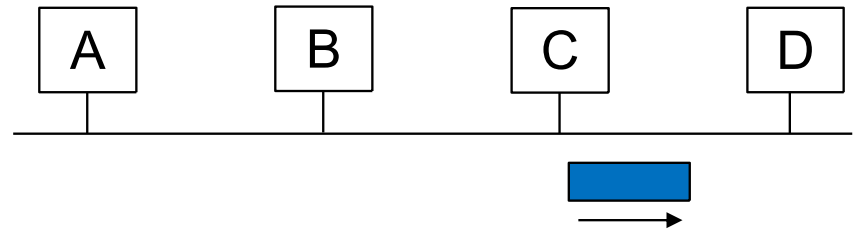
# Collisions in Bus Topology Ethernet

- ❖ Collision may happen in Ethernet of bus topology.



- ❖ For example:

- A sends a frame at time  $t$ .
- A's frame reaches D at time  $t + d$ .
- D begins transmission at time  $t + d - 1$  and collides with A's frame.



# Ethernet CSMA/CD Algorithm

1. NIC receives datagram from network layer, creates frame.
2. If NIC senses channel idle, starts frame transmission. If NIC senses channel busy, waits until channel idle, then transmits.
3. If NIC transmits entire frame without detecting another transmission, NIC is done with frame!
4. If NIC detects another transmission while transmitting, aborts and sends jam signal.
5. After aborting, NIC enters binary back-off:
  - after  $m^{th}$  collision, NIC chooses  $K$  at random from  $\{0, 1, 2, \dots, 2^m - 1\}$ .
  - NIC waits  $K * 512$  bit times, returns to Step 2.



# Ethernet CSMA/CD Algorithm

## Exponential backoff:

- ❖ After 1<sup>st</sup> collision: choose  $K$  at random from  $\{0, 1\}$ ; wait  $K * 512$  bit transmission times before retransmission.
- ❖ After 2<sup>nd</sup> collision: choose  $K$  from  $\{0, 1, \dots, 2^2 - 1\}$ .
- ...
- ❖ After  $m^{th}$  collision, choose  $K$  at random from  $\{0, 1, \dots, 2^m - 1\}$
- ❖ *Goal*: adapt retransmission attempts to estimated current load
  - More collisions implies heavier load.
  - longer back-off interval with more collisions.

# Lectures 8&9: 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

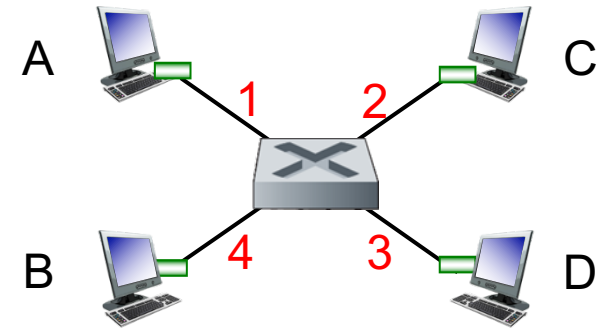
- ❖ A link-layer device used in LAN
  - Store and forward Ethernet frames
  - Examine incoming frame's MAC address, selectively forward frame to one-or-more outgoing links.
- ❖ Transparent to hosts
  - No IP address
  - Hosts are unaware of the presence of switches



a 50-port Ethernet switch  
(Source: Wikipedia)

# Ethernet Switch

- ❖ In Ethernet of star topology, hosts have dedicated connection to switch.
- ❖ Switch buffers frames and is full duplex.
  - A and D can send frames to each other simultaneously.
- ❖ Ethernet protocol is used on each link, but no collisions!



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

# Switch Forwarding Table

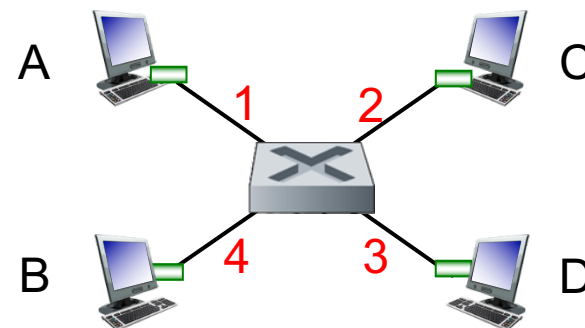
❖ **Q:** how does switch know A is reachable via interface 1, B is reachable via interface 4?

❖ **A:** each switch has a **switch table**.

- Format of entry:

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

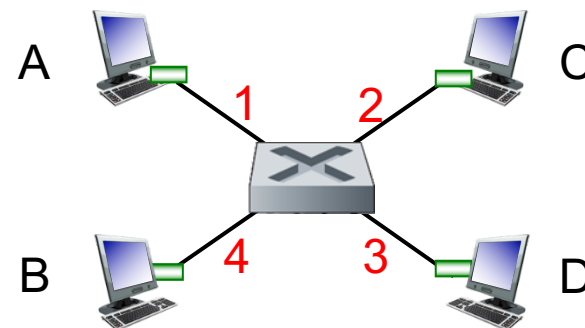
❖ **Q:** how are entries created and maintained in a switch table?



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

# Switch: Self-learning

- ❖ Switch *learns* which hosts can be reached through which interfaces.
  - When receiving a frame from *A*, note down the location of *A* in switch table.
  - If destination *B* is **found** in the table, forward the frame onto that link.
  - If destination *B* is **unknown**, broadcast the frame to all outgoing links.



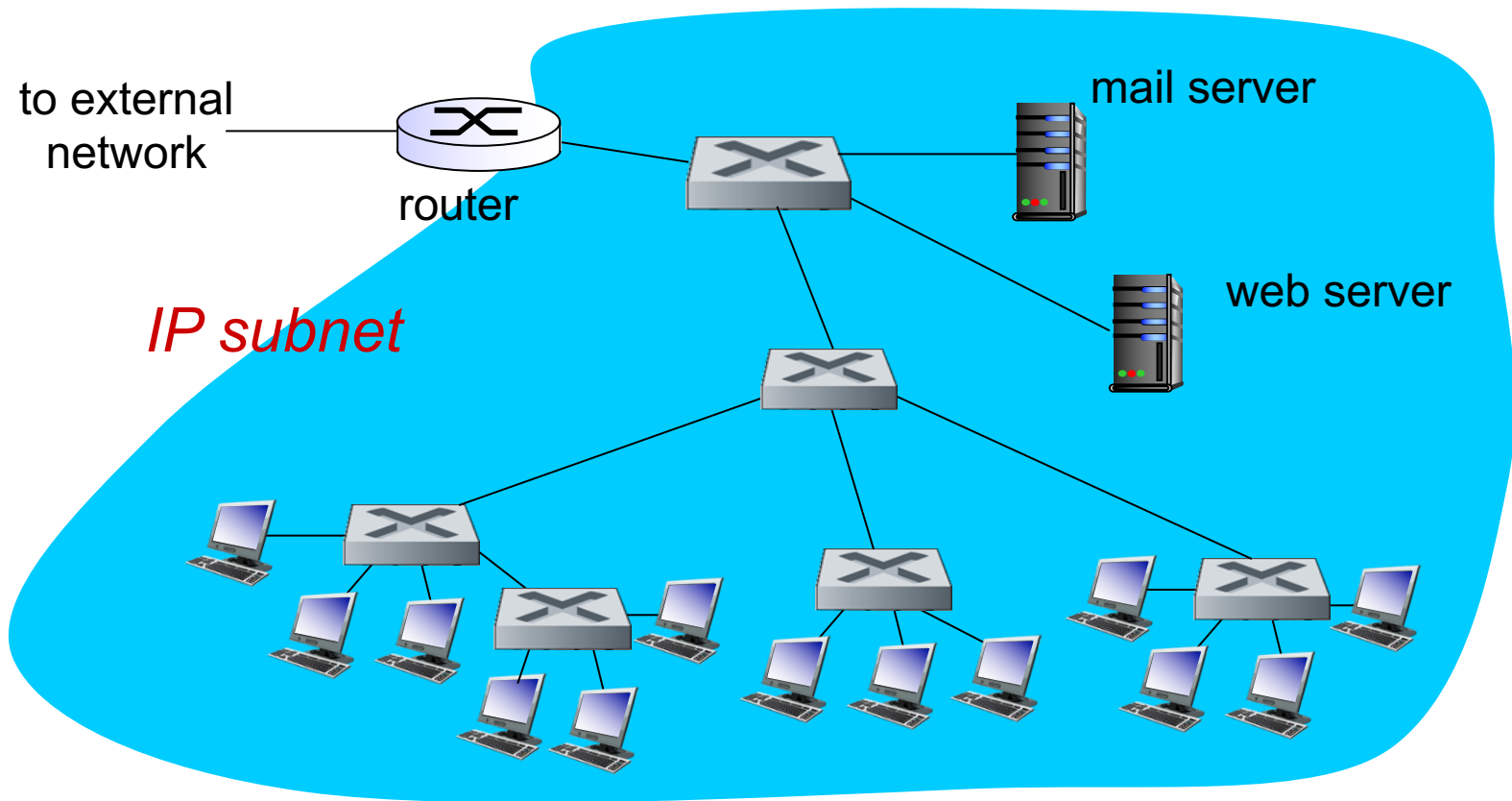
A switch with 4 interfaces  
(1, 2, 3, 4)

MAC addr	Interface	TTL
A	1	60

Switch table (initially empty)

# Interconnecting Switches

- ❖ Switches can be connected in hierarchy.



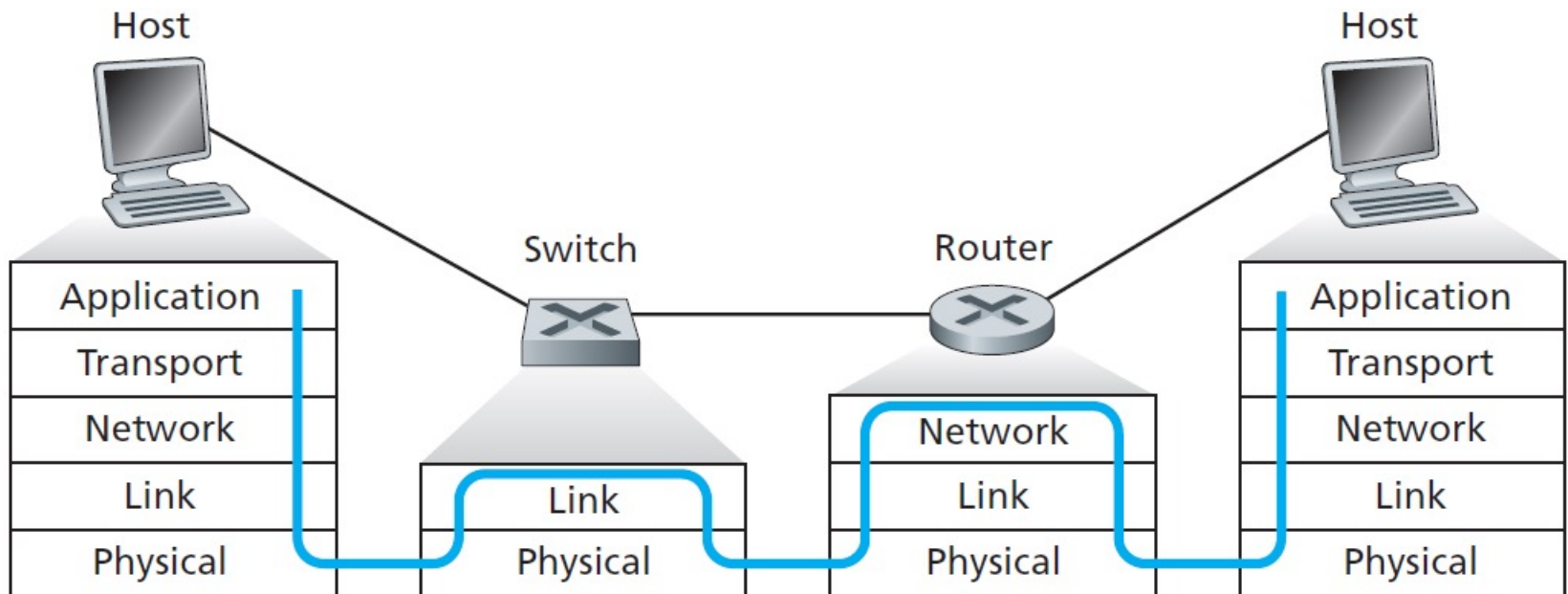
# Switches vs. Routers

## ❖ 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





# Lecture 9: 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
  - CSMA/CD protocol with binary back-off
  - Ethernet switches and switch tables