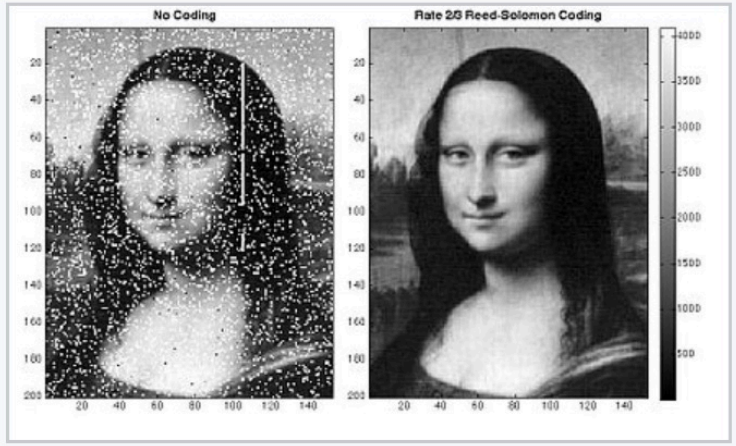
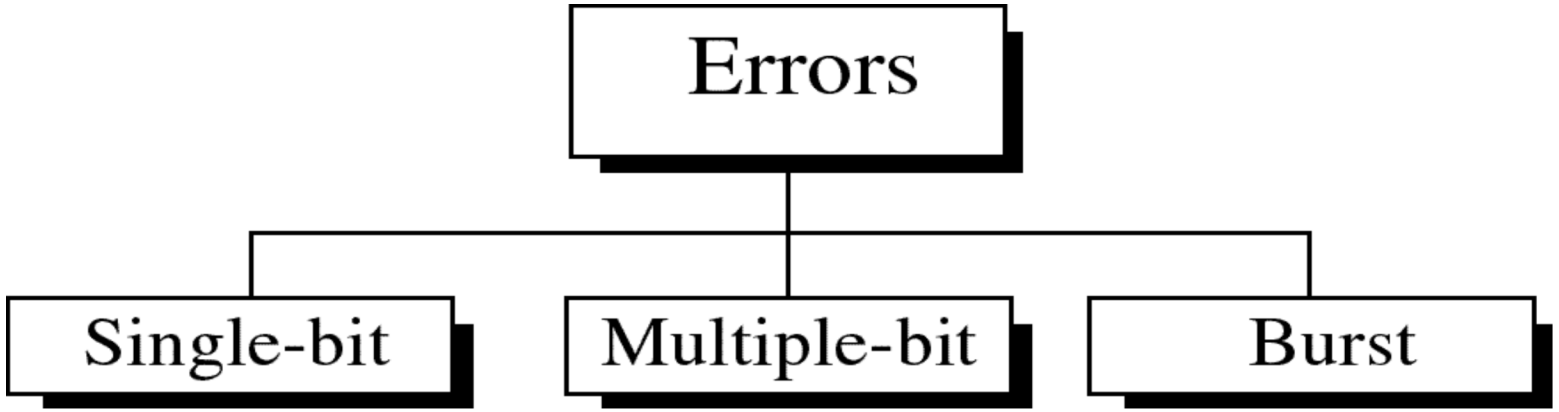
Error Detection and Correction

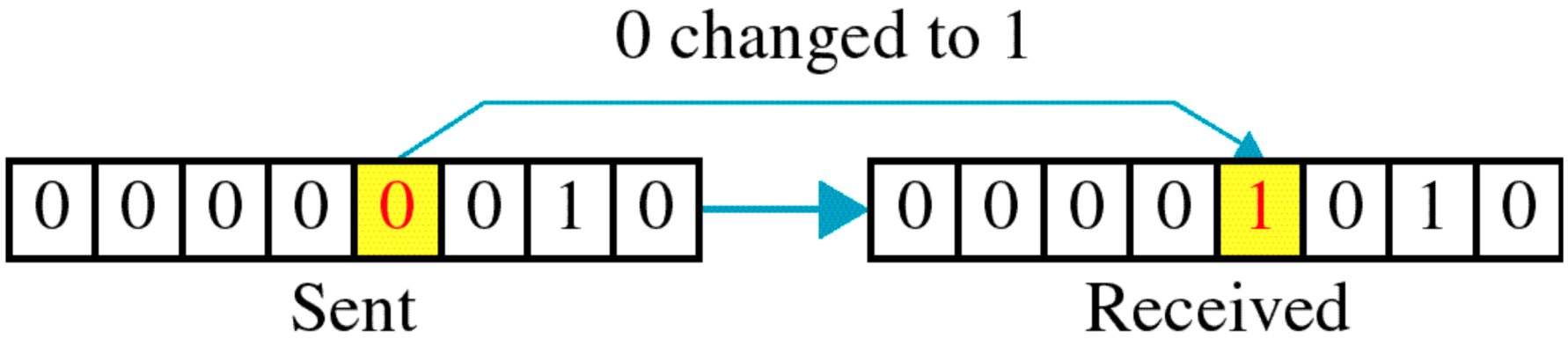
Error Types



To clean up transmission errors introduced by Earth's atmosphere (left), Goddard scientists applied Reed–Solomon error correction (right), which is commonly used in CDs and DVDs. Typical errors include missing pixels (white) and false signals (black). The white stripe indicates a brief period when transmission was interrupted.



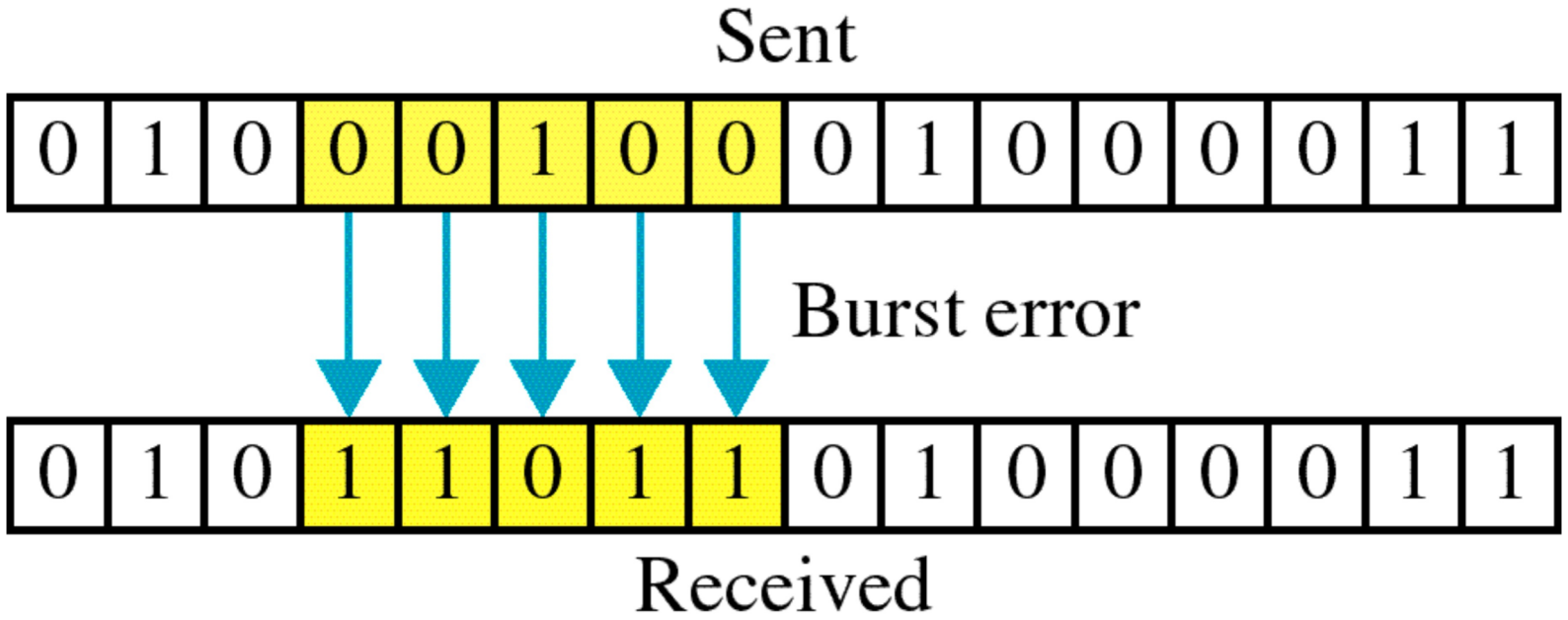
Single Bit Error



Single bit errors are the least likely type of errors in serial data transmission because the noise must have a very short duration which is very rare.

- If data is sent at 1Mbps then each bit lasts only 1/1,000,000 sec. or 1 µs.
- For a single-bit error to occur, the noise must have a duration of only 1 µs, which is very rare.

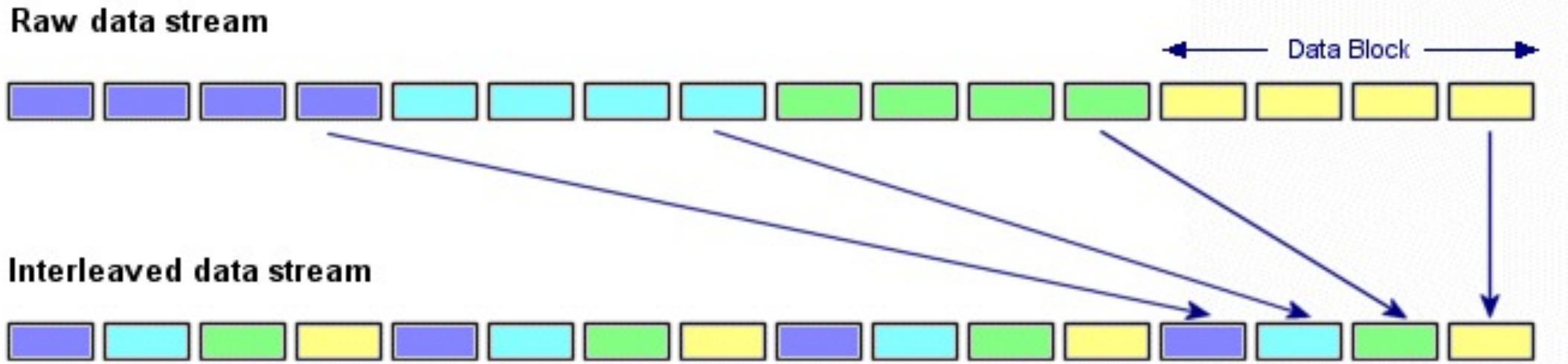
Burst Errors

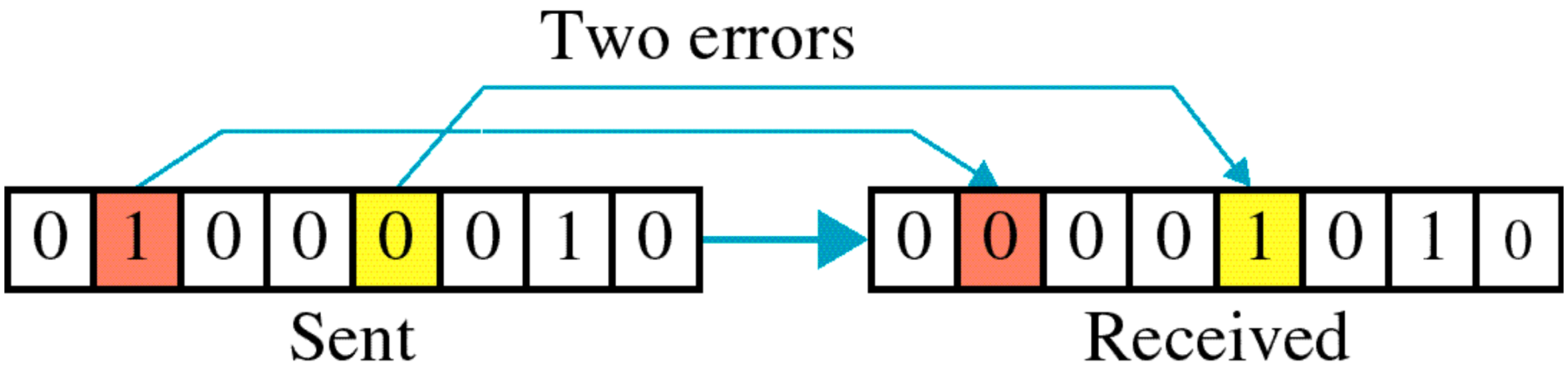


The term burst error means that two or more bits in the data unit have changed from 1 to 0 or from 0 to 1.

- If data is sent at rate = 1Kbps then a noise of 1/100 sec can affect 10 bits.(1/100*1000)
- If same data is sent at rate = 1Mbps then a noise of 1/100 sec can affect 10,000 bits.(1/100*10^6)

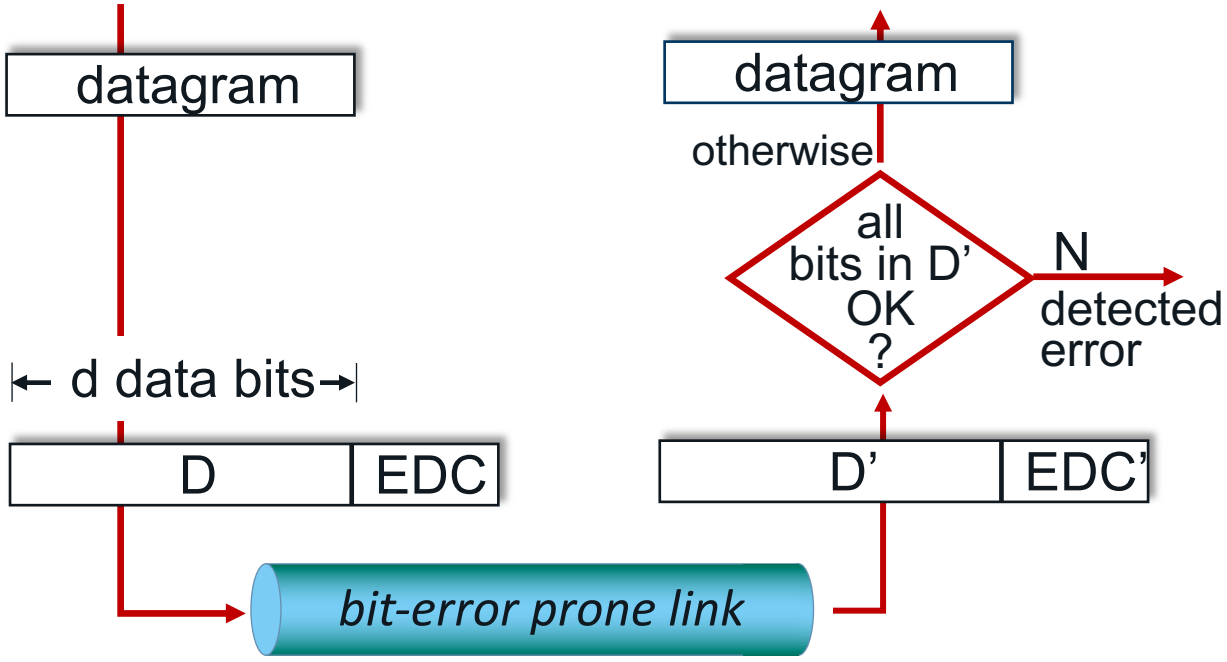
Interleaving against Burst Errors





Error Detection

EDC: error detection and correction bits (e.g., redundancy)
D: data protected by error checking, may include header fields



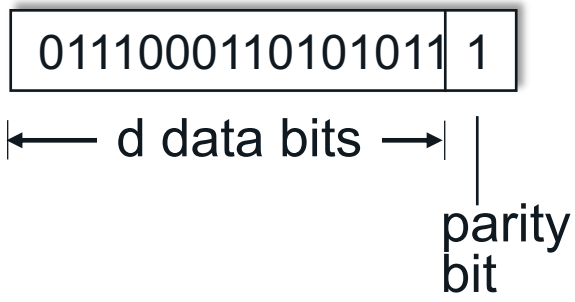
Error detection not 100% reliable!

- protocol may miss some errors, but rarely
- larger EDC field yields better detection and correction

Error Correction

single bit parity:

- detect single bit errors

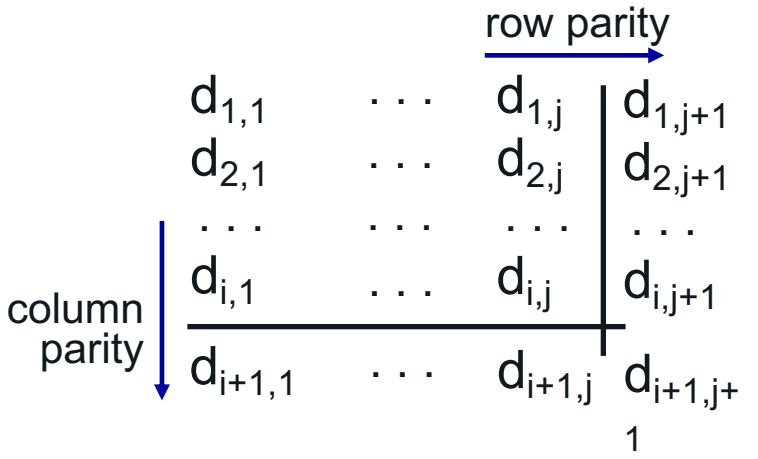


Even parity: set parity bit so there is an even number of 1's

Can detect odd number of bit errors.

two-dimensional bit parity:

- detect *and correct* single bit errors



no errors:

1	0	1	0	1		1
1	1	1	1	0		0
0	1	1	1	0		1
0	0	1	0	1		0

detected and correctable single-bit error:

1	0	1	0	1		1
1	0	1	1	0		0
0	1	1	1	0		1
0	0	1	0	1		0

parity error

parity error

Example

Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer be at the rghit pclae. The rset can be a toatl mses and you can sitll raed it wouthit porbelm. Tihs is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe.

Where is the redundancy?

Thank You!


Tell me and I
forget. Teach me
and I remember.
Involve me and I
learn.
~ Benjamin Franklin

Multiple Access

Multiple Access: Links and Protocols

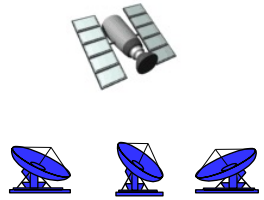
two types of “links”:

- point-to-point
  - point-to-point link between Ethernet switch, host
  - PPP for dial-up access
- broadcast (shared wire or medium)
  - old-fashioned Ethernet
  - upstream HFC in cable-based access network
  - 802.11 wireless LAN, 4G/4G. satellite

shared wire (e.g., cabled Ethernet)

shared radio: 4G/5G

shared radio: WiFi



shared radio: satellite

humans at a cocktail party (shared air, acoustical)

Multiple Access: Links and Protocols

- single shared broadcast channel
- two or more simultaneous transmissions by nodes: interference
  - *collision* if node receives two or more signals at the same time

multiple access protocol

- distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
- communication about channel sharing must use channel itself!
  - no out-of-band channel for coordination

An Ideal Multiple Access Scenario

given: multiple access channel (MAC) of rate *R* bps

desiderata:

1. when one node wants to transmit, it can send at rate *R*.
2. when M nodes want to transmit, each can send at average rate *R/M*
3. fully decentralized:
   - no special node to coordinate transmissions
   - no synchronization of clocks, slots
4. simple

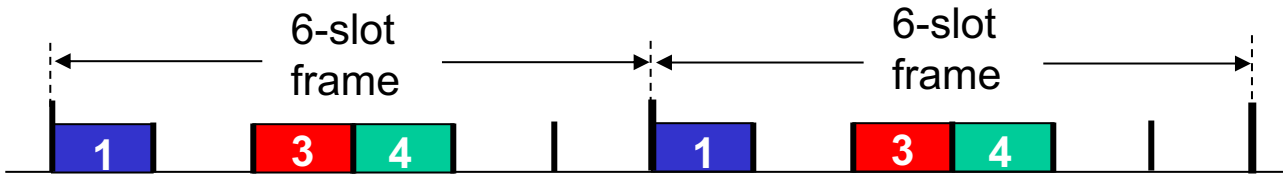
Three Multiple Access Types

- channel partitioning
  - divide channel into smaller “pieces” (time slots, frequency, code)
  - allocate piece to node for exclusive use
- *random access*
  - channel not divided, allow collisions
  - “recover” from collisions
- “taking turns”
  - nodes take turns, but nodes with more to send can take longer turns

Channel Partitioning Multiple Access: TDMA

TDMA: time division multiple access

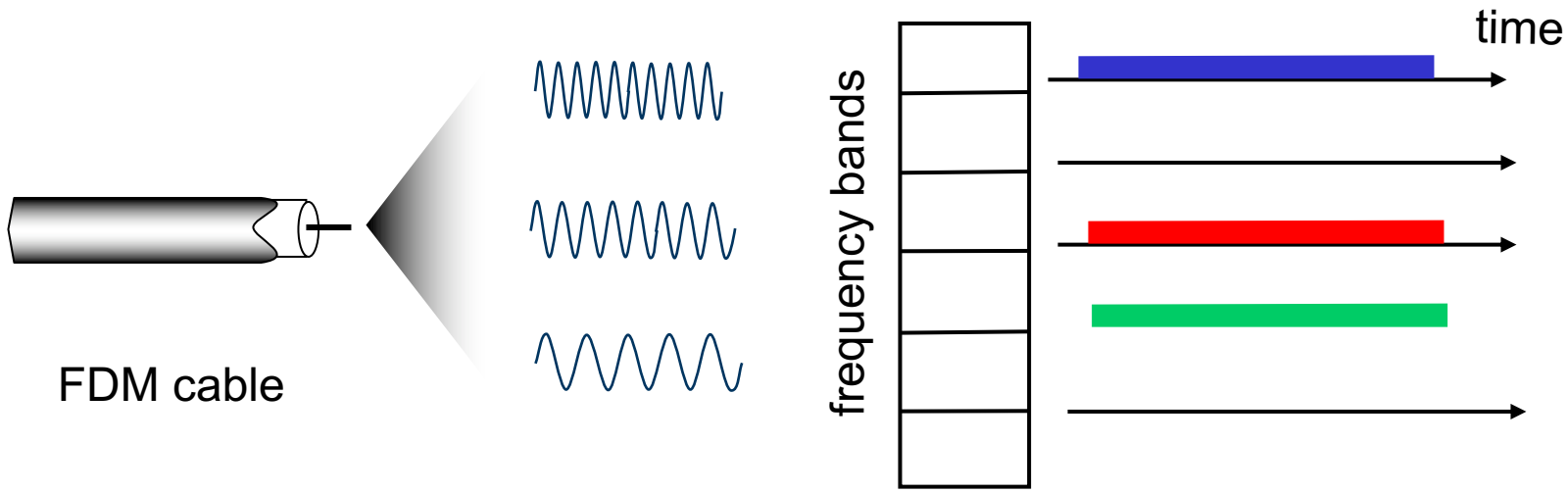
- access to channel in “rounds”
- each station gets fixed length slot (length = packet transmission time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have packets to send, slots 2,5,6 idle



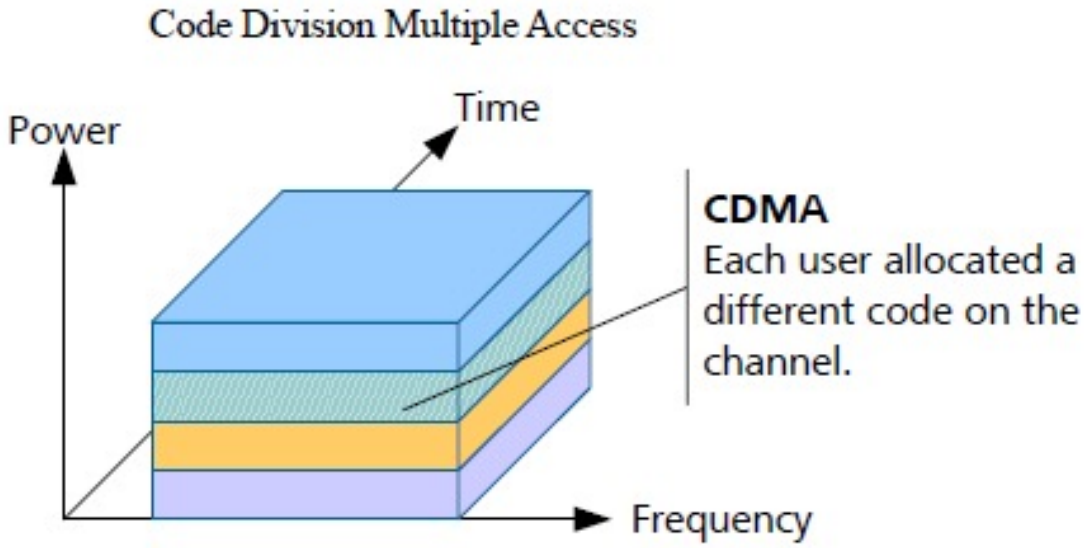
Channel Partitioning Multiple Access: FDMA

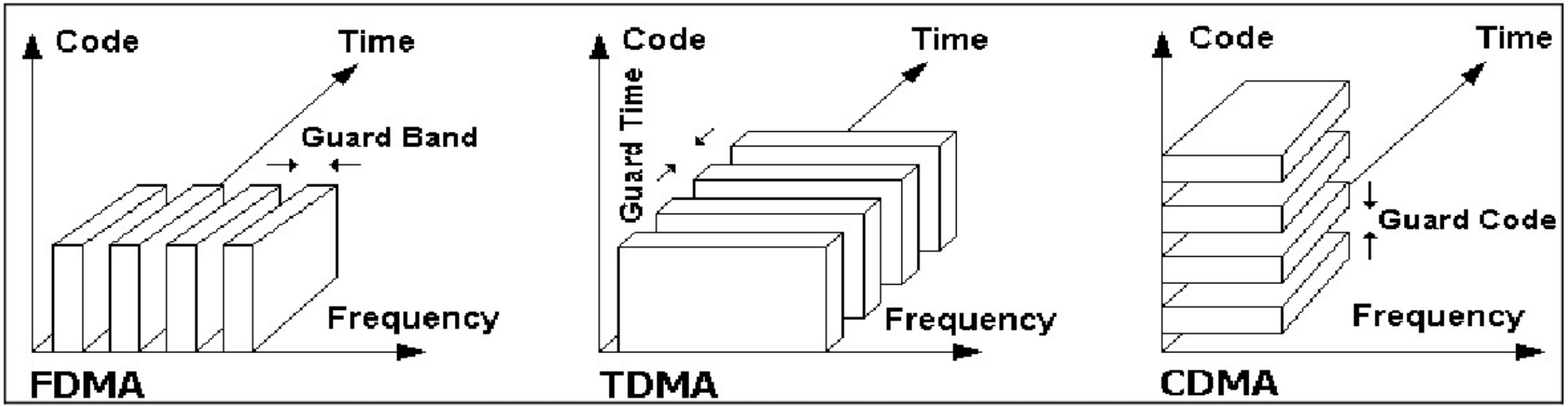
FDMA: frequency division multiple access

- channel spectrum divided into frequency bands
- each station assigned fixed frequency band
- unused transmission time in frequency bands go idle
- example: 6-station LAN, 1,3,4 have packet to send, frequency bands 2,5,6 idle

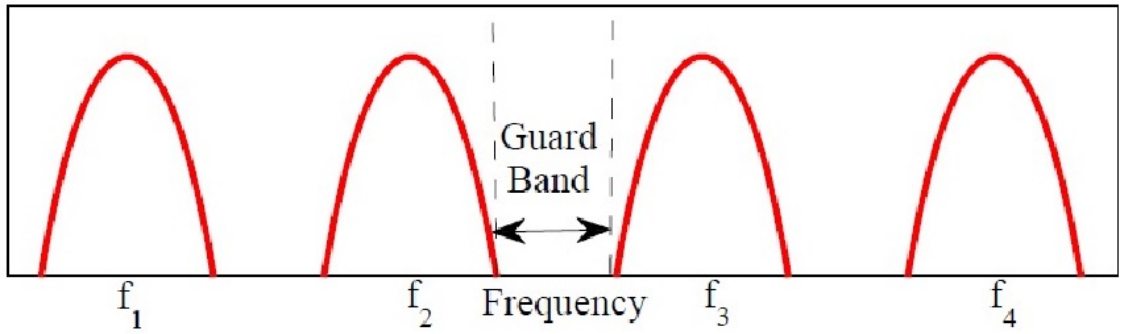


Channel Partitioning Multiple Access: CDMA

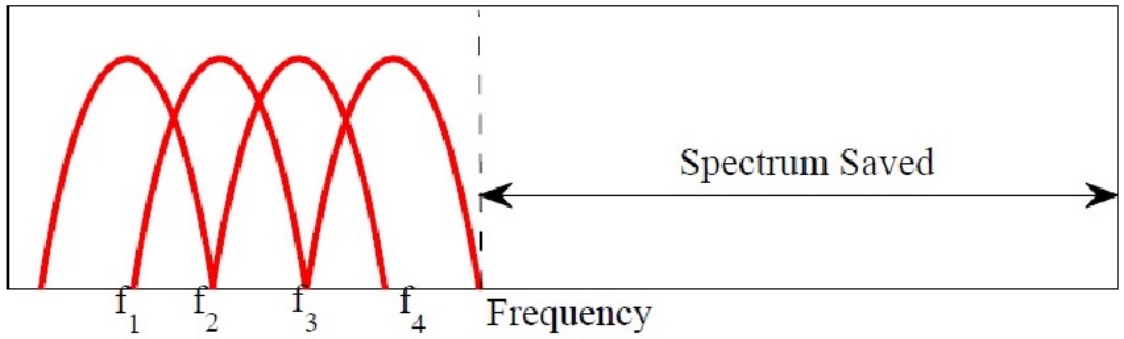




Channel Partitioning Multiple Access: OFDMA



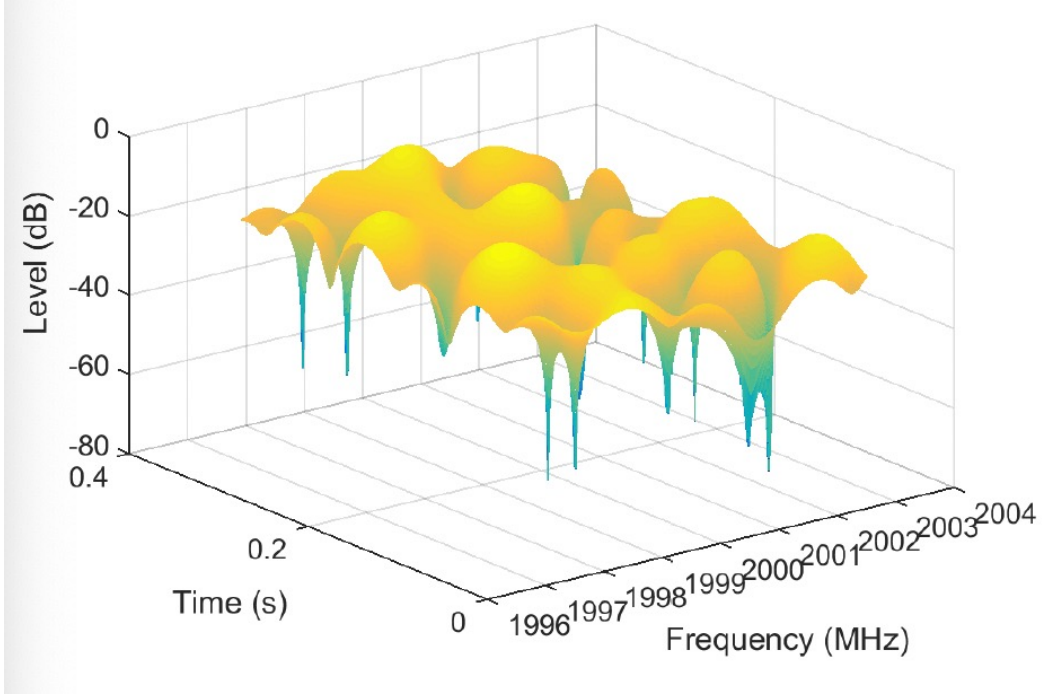
(a) Conventional FDM.



(b) Orthogonal FDM.

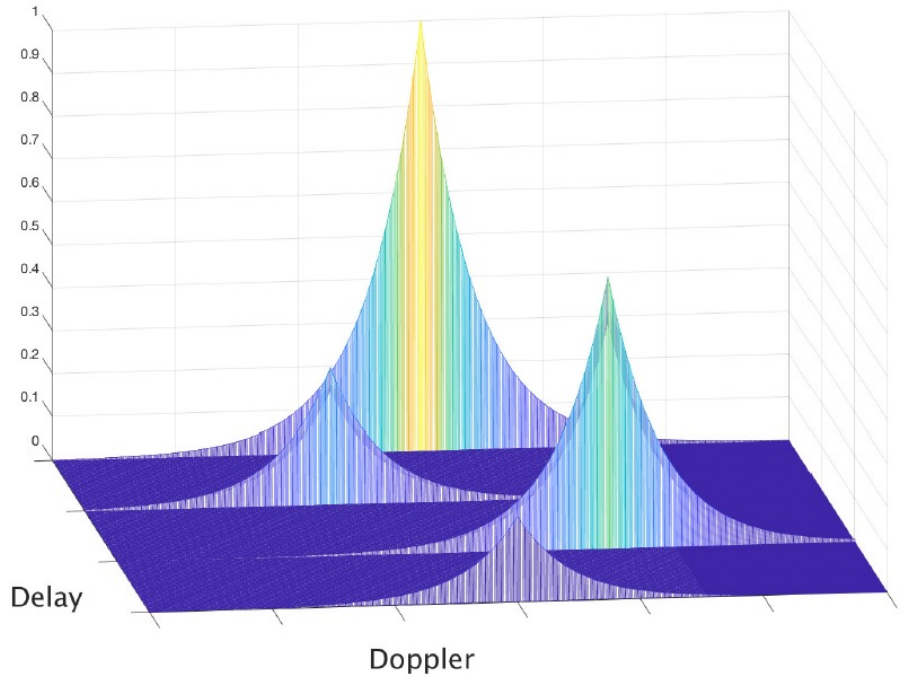
Generation	Technology	Feature	Encoding	Year of First Use
1G	FDMA	NMT	Analog	1981
2G	TDMA and FDMA	GSM	Digital	1991
2G	CDMA	IS-95 (CDMA one)	Digital	1995
3G	CDMA	IS-2000 (CDMA 2000)	Digital	2000 / 2002
3G	W-CDMA	UMTS (3GSM)	Digital	2001
4G	OFDMA	LTE	Digital	2009
5G	OFDMA	NR	Digital	2018

Channel Partitioning Multiple Access: OTFS



SFFT

ISFFT



“Taking Turn” Multiple Access

channel partitioning MAC protocols:

- share channel *efficiently* and *fairly* at high load
- inefficient at low load: delay in channel access, 1/N bandwidth allocated even if only 1 active node!

“taking turns” protocols

- look for best of both worlds!

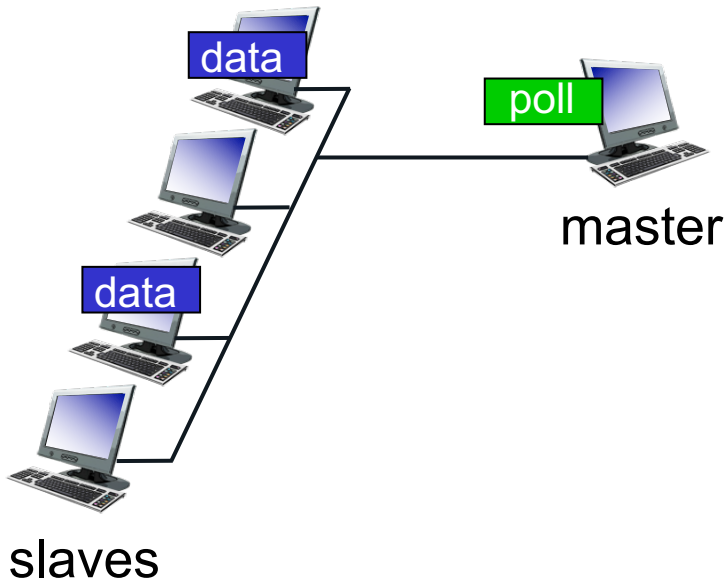
random access MAC protocols

- efficient at low load: single node can fully utilize channel
- high load: collision overhead

“Taking Turn” Multiple Access

polling:

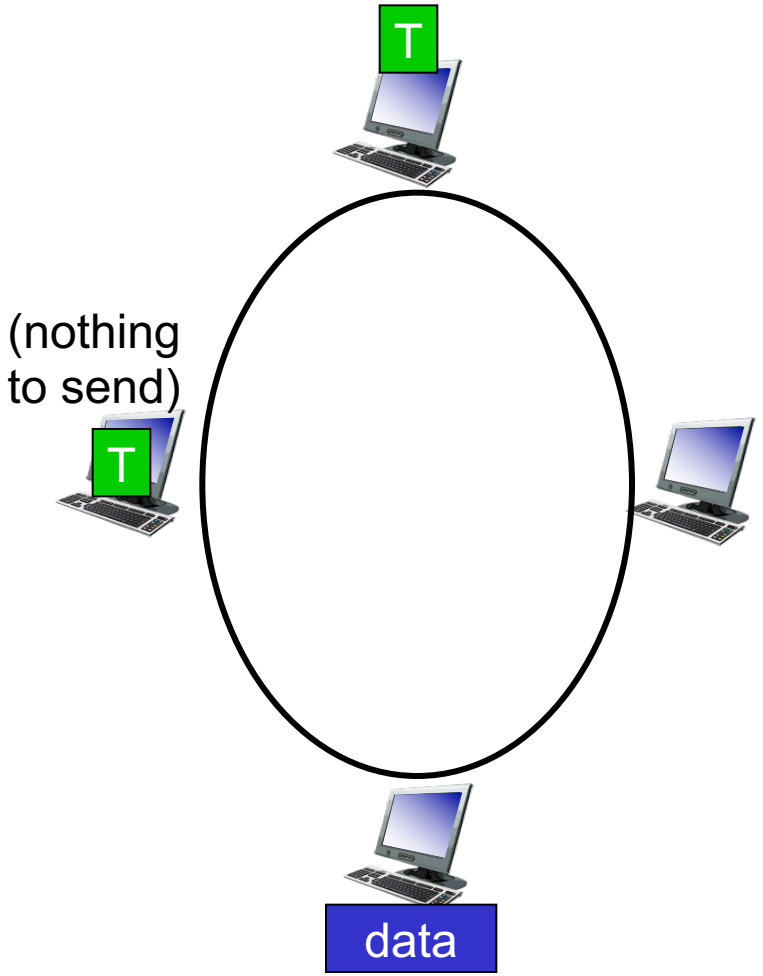
- master node “invites” other nodes to transmit in turn
- typically used with “dumb” devices
- concerns:
  - polling overhead
  - latency
  - single point of failure (master)



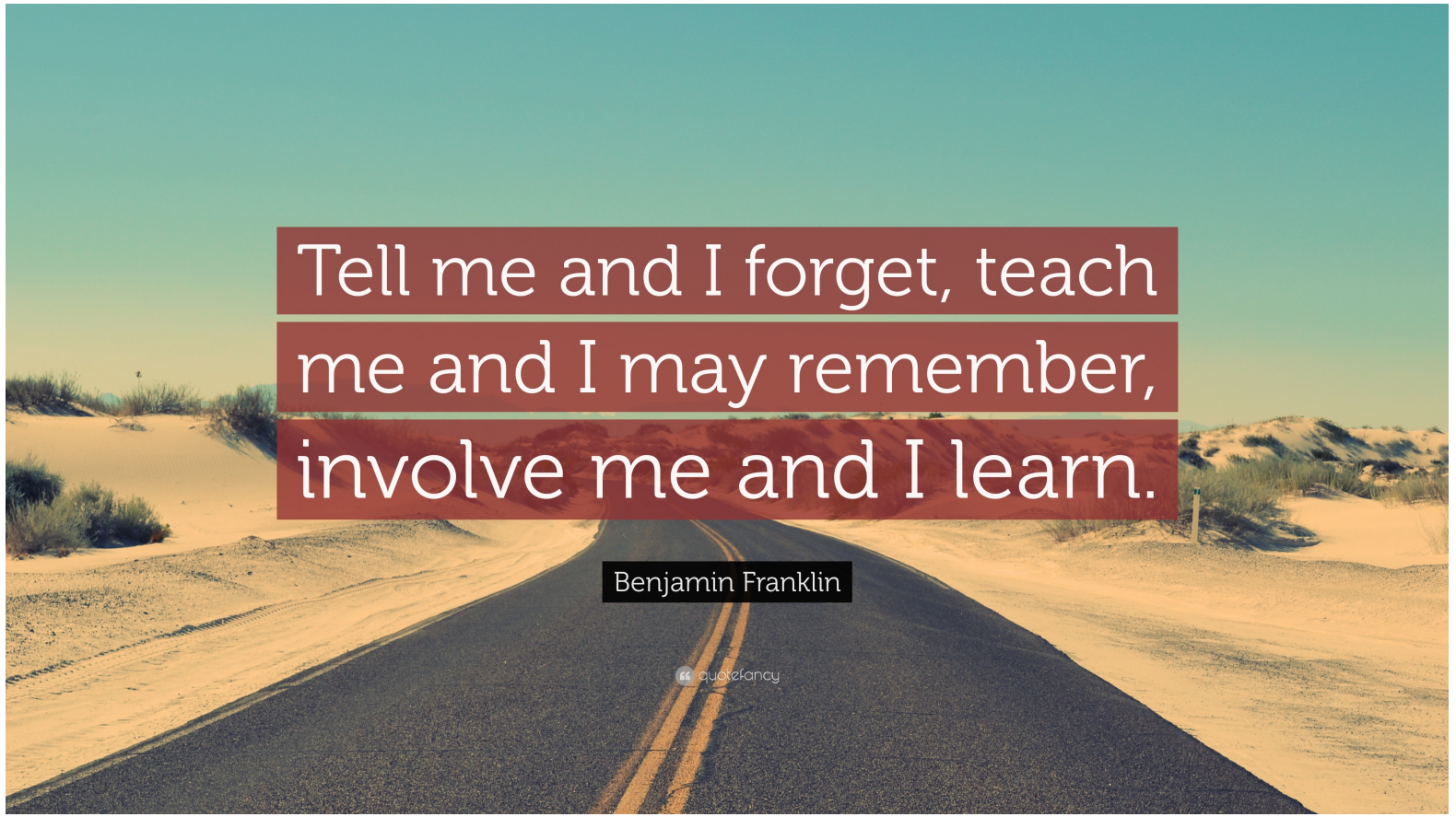
“Taking Turn” Multiple Access

token passing:

- control *token* passed from one node to next sequentially.
- token message
- concerns:
  - token overhead
  - latency
  - single point of failure (token)



Thank You!


Tell me and I forget, teach
me and I may remember,
involve me and I learn.
Benjamin Franklin
quotefancy

Random Multiple Access

Random Multiple Access

channel partitioning MAC protocols:

- share channel *efficiently* and *fairly* at high load
- inefficient at low load: delay in channel access, 1/N bandwidth allocated even if only 1 active node!

"taking turns" protocols

- look for best of both worlds!

random access MAC protocols

- efficient at low load: single node can fully utilize channel
- high load: collision overhead

Slotted ALOHA

assumptions:

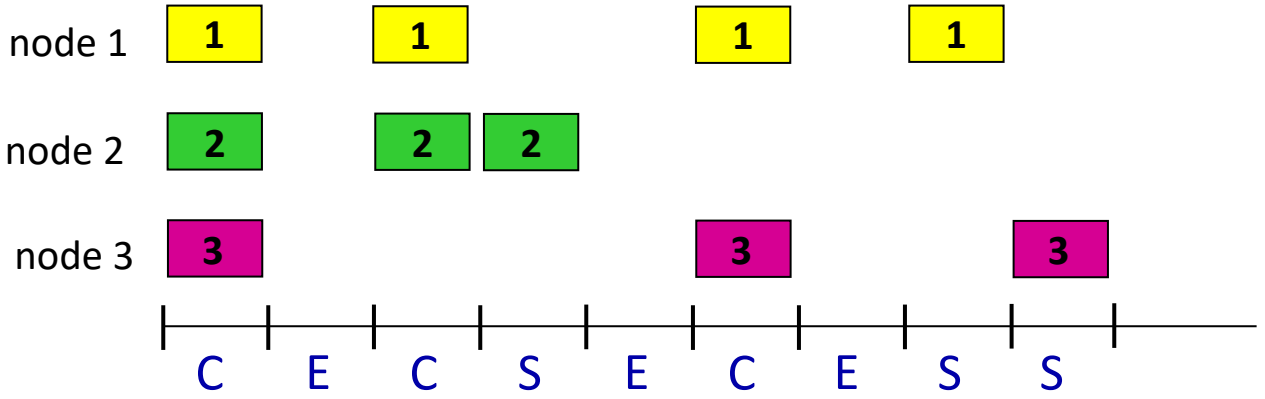
- all frames same size
- time divided into equal size slots (time to transmit 1 frame)
- nodes start to transmit only slot beginning
- nodes are synchronized
- if 2 or more nodes transmit in slot, all nodes detect collision

operation:

- when node obtains fresh frame, transmits in next slot
  - *if no collision:* node can send new frame in next slot
  - *if collision:* node retransmits frame in each subsequent slot with probability *p* until success

randomization – *why*?

Slotted ALOHA



C: collision
S: success
E: empty

Pros:

- single active node can continuously transmit at full rate of channel
- highly decentralized: only slots in nodes need to be in sync
- simple

Cons:

- collisions, wasting slots
- idle slots
- nodes may be able to detect collision in less than time to transmit packet
- clock synchronization

Slotted ALOHA: Efficiency

efficiency: long-run fraction of successful slots (many nodes, all with many frames to send)

suppose: N nodes with many frames to send, each transmits in slot with probability p

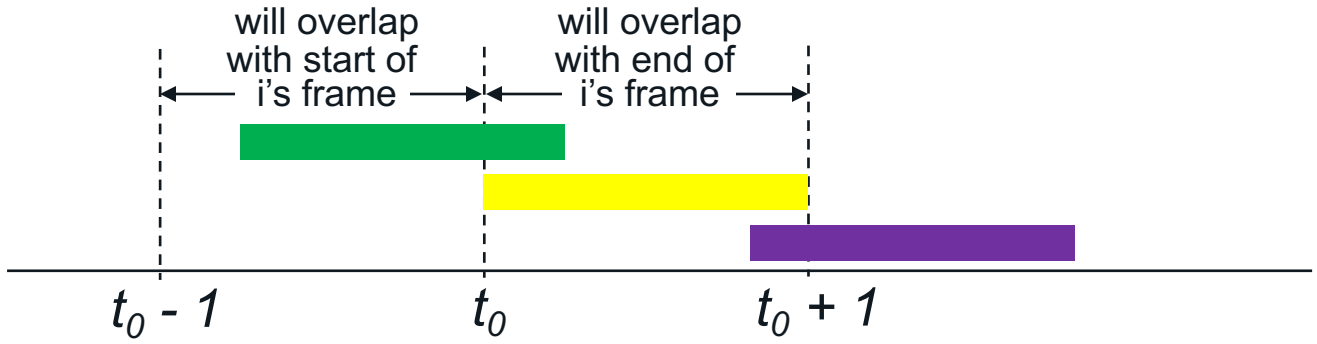
- prob that given node has success in a slot = $p(1-p)^{N-1}$
- prob that *any* node has a success = $Np(1-p)^{N-1}$
- max efficiency: find p^* that maximizes $Np(1-p)^{N-1}$
- for many nodes, take limit of $Np^*(1-p^*)^{N-1}$ as N goes to infinity, gives:

max efficiency = 1/e = .37

at best: channel used for useful transmissions 37% of time!

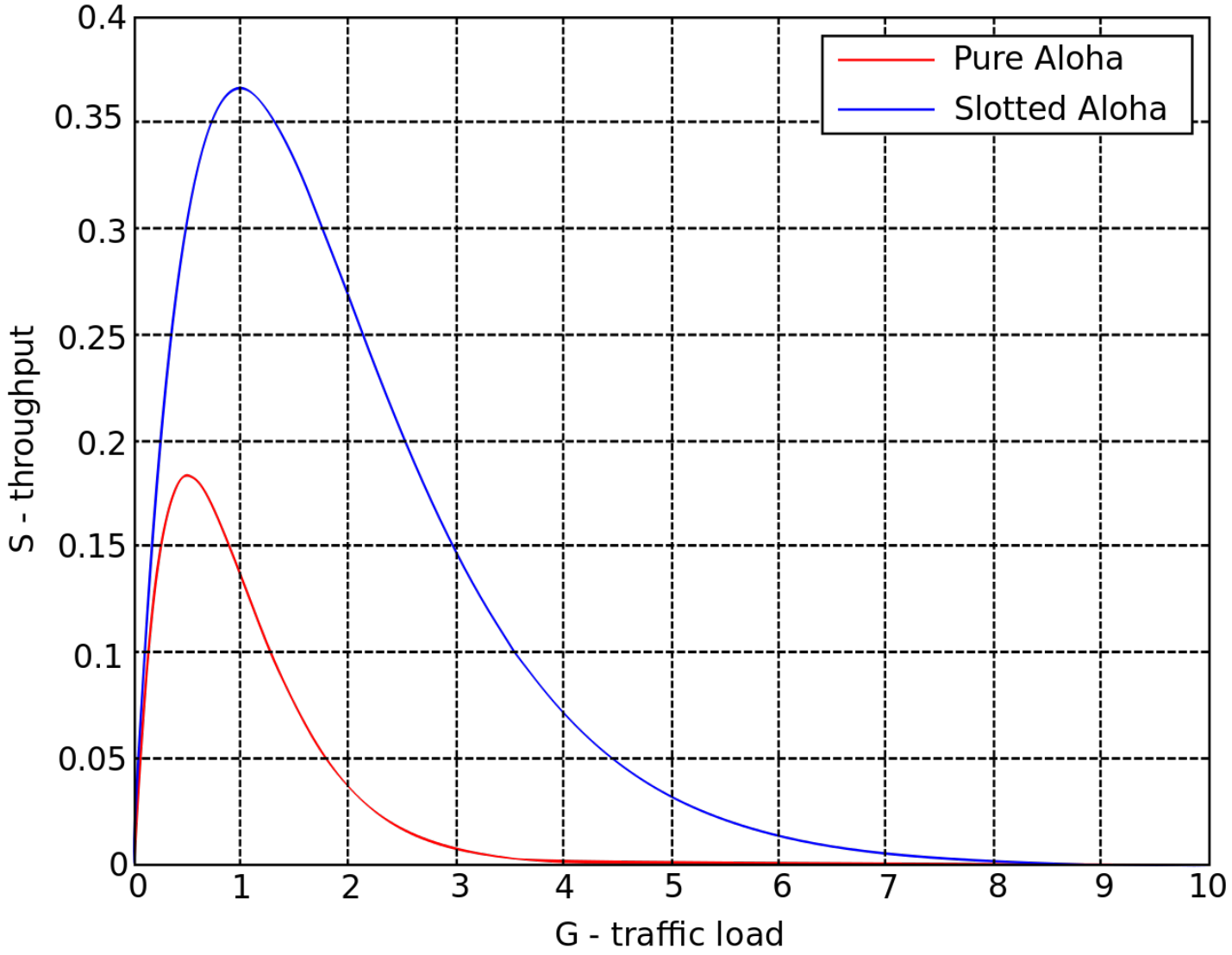
Pure ALOHA

- unslotted Aloha: simpler, no synchronization
  - when frame first arrives: transmit immediately
- collision probability increases with no synchronization:
  - frame sent at t_0 collides with other frames sent in $[t_0-1,t_0+1]$



- pure Aloha efficiency: 18% !

ALOHA Efficiency



CSMA (Carrier Sense Multiple Access)

simple CSMA: listen before transmit:

- if channel sensed idle: transmit entire frame
- if channel sensed busy: defer transmission

- human analogy: don't interrupt others!

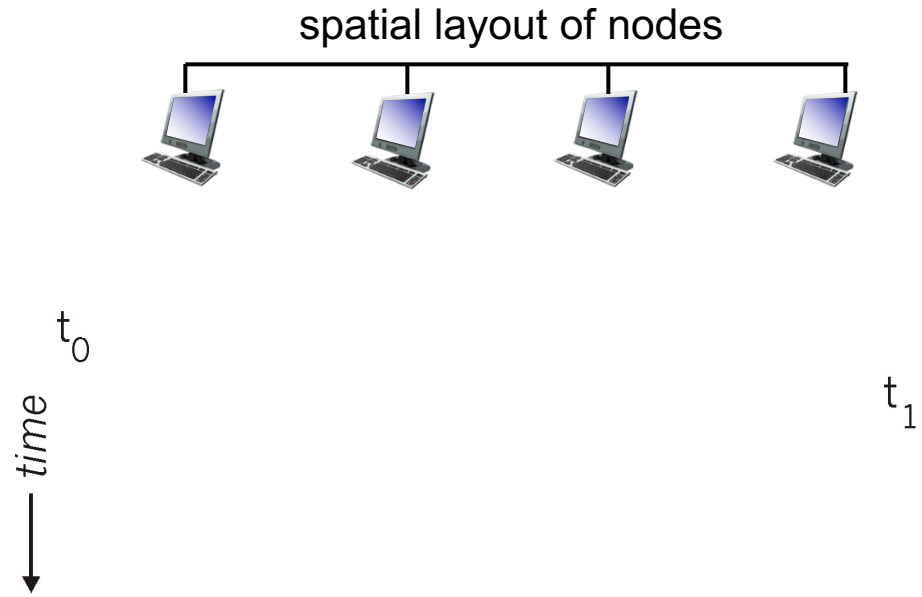
CSMA/CD: CSMA with *collision detection*

- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage
- collision detection easy in wired, difficult with wireless

- human analogy: the polite conversationalist

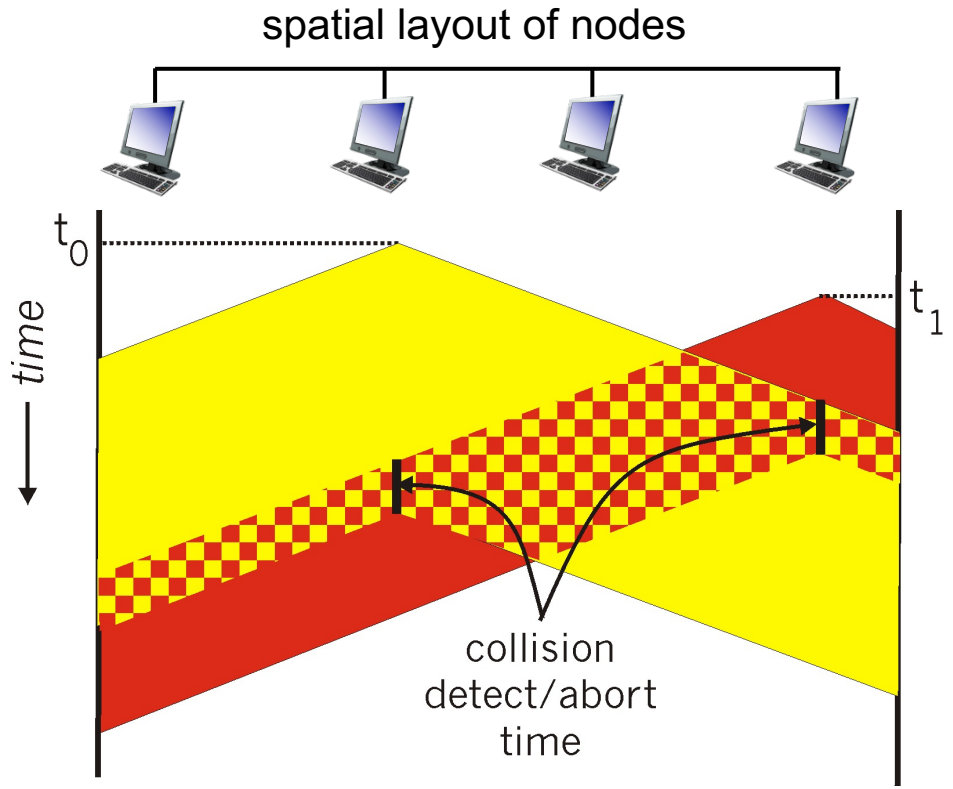
CSMA: Collisions

- collisions *can* still occur with carrier sensing:
  - propagation delay means two nodes may not hear each other's just-started transmission
- collision: entire packet transmission time wasted
  - distance & propagation delay play role in in determining collision probability



CSMA/CD

- CSMA/CS reduces the amount of time wasted in collisions
  - transmission aborted on collision detection



Ethernet CSMA/CD Algorithm

1. NIC receives datagram from network layer, creates frame
2. If NIC senses channel:

   if idle: start frame transmission.

   if busy: wait until channel idle, then transmit
3. If NIC transmits entire frame without collision, NIC is done with frame !
4. If NIC detects another transmission while sending: abort, send jam signal
5. After aborting, NIC enters *binary (exponential) backoff:*
   - after *m*th collision, NIC chooses *K* at random from $\{0,1,2, \ldots, 2^m-1\}$. NIC waits $K \cdot 512$ bit times, returns to Step 2
   - more collisions: longer backoff interval

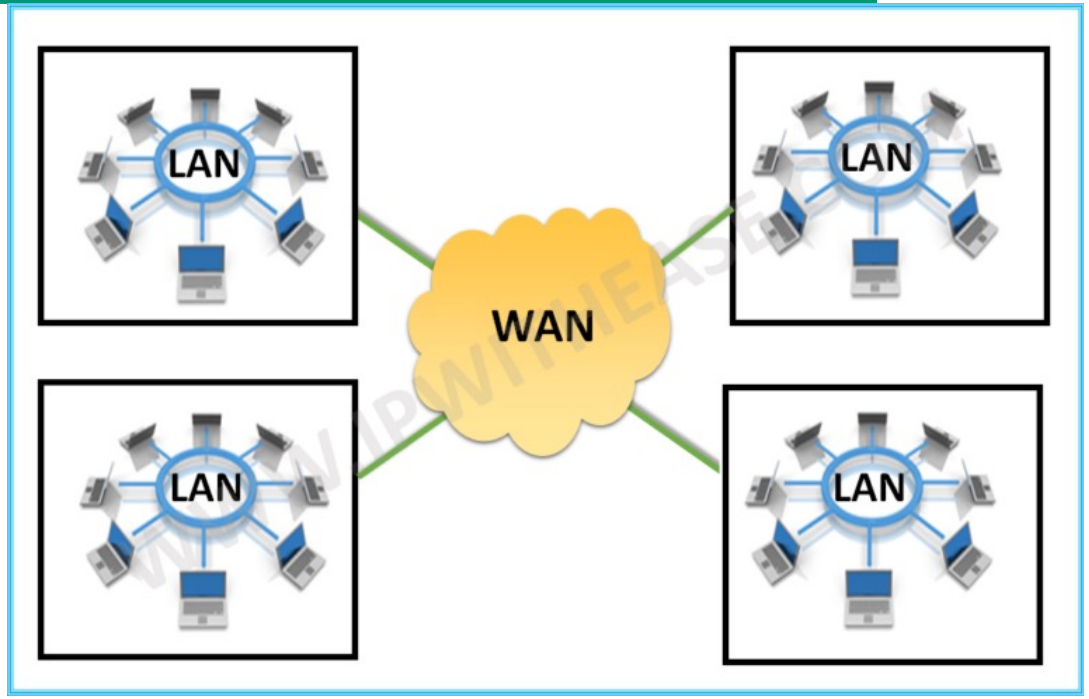
Thank You!


Tell me and I
forget. Teach me
and I remember.
Involve me and I
learn.
~ Benjamin Franklin

LAN and WAN

LAN and WAN

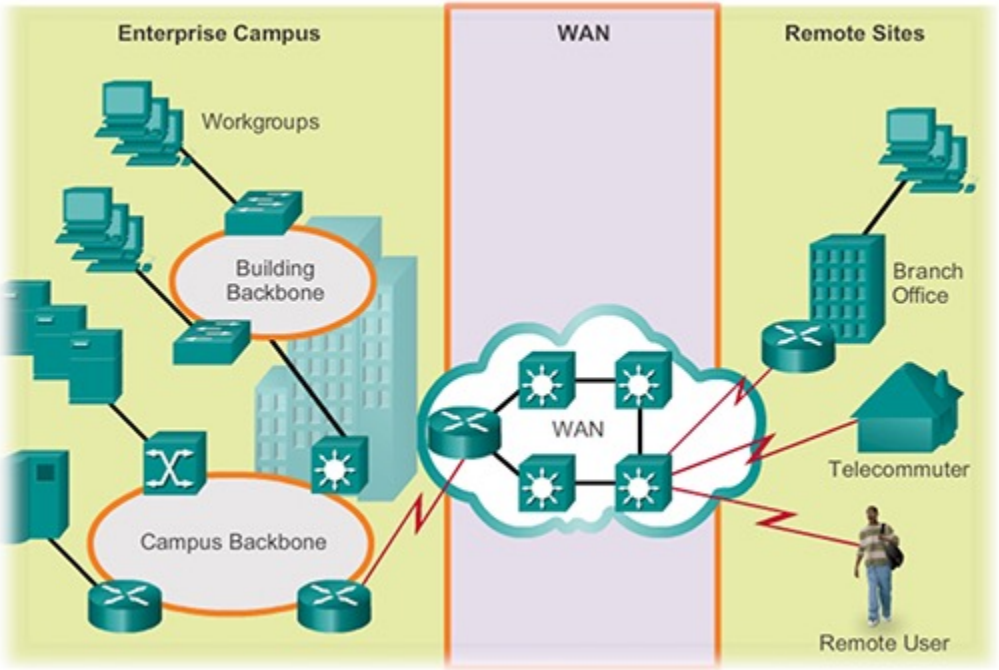
A LAN (local area network) is a group of computers and network devices connected together, usually within the same building. By definition, the connections must be high speed and relatively inexpensive (e.g., token ring or Ethernet).



A WAN (wide area network) is not restricted to a geographical location, although it might be confined within the bounds of a state or country. A WAN connects several LANs, and may be limited to an enterprise (a corporation or an organization) or accessible to the public. The technology is high speed and relatively expensive. The Internet is an example of a worldwide public WAN.

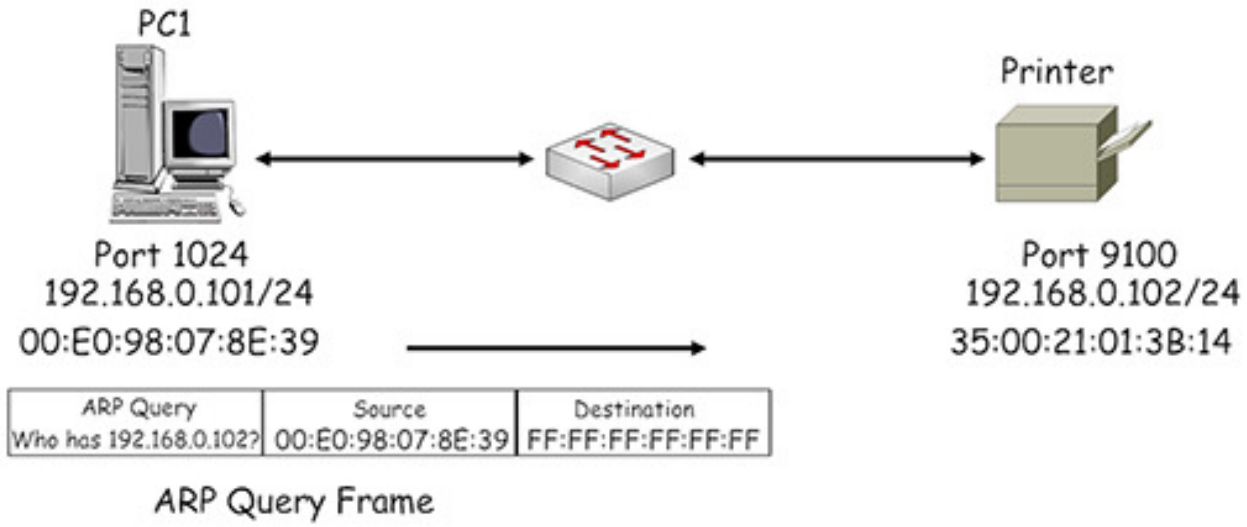
LAN and WAN

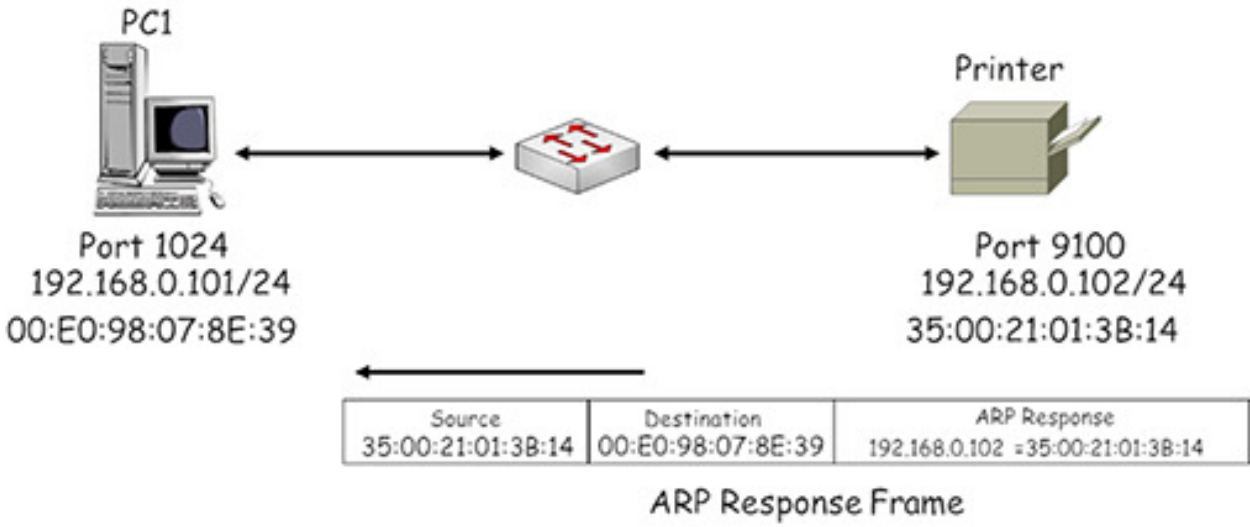
At home, your Local Area Network (LAN) might connect together devices over a distance measured in tens of metres. At work or school, the LAN might connect devices over hundreds of metres. A Wide Area Network (WAN) operates over a much larger area, as they interconnect LANs to allow them to exchange data.



For example, in the diagram above a large business network needs to provide a connection to a remote branch office, and to employees who work from home (telecommuters) and who are travelling (remote users). This connection is provided by a WAN. WANs are operated by a service provider, and businesses pay them a fee in order to gain access.

Sending data in the LAN





PC1 is unable to identify the correct MAC address to place in the destination field of the frame, so the frame cannot be transmitted to the printer. Consequently PC1 initiates a broadcast communication to all the devices in its IP subnet using the Address Resolution Protocol (ARP), requesting information about the MAC address associated with the device using IP address 192.168.0.102.

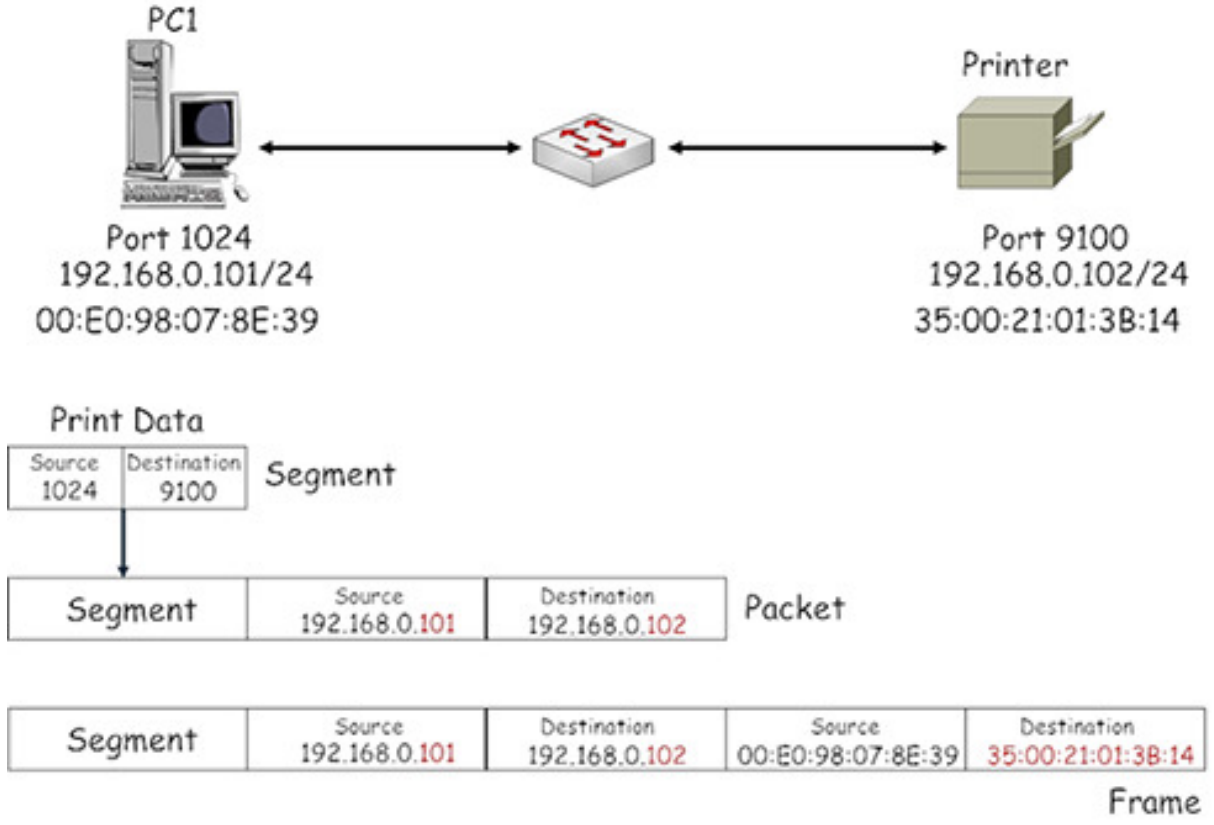
Note the destination address used by the ARP is FF:FF:FF:FF:FF:FF – this is a broadcast address, which is flooded by the Ethernet switch from all its ports (apart from the one the address was received on). Thus, all the devices within the LAN receive the ARP query, but only the printer responds as the query contains its IP address. It returns an ARP response, identifying its assigned MAC address:

Sending data in the LAN

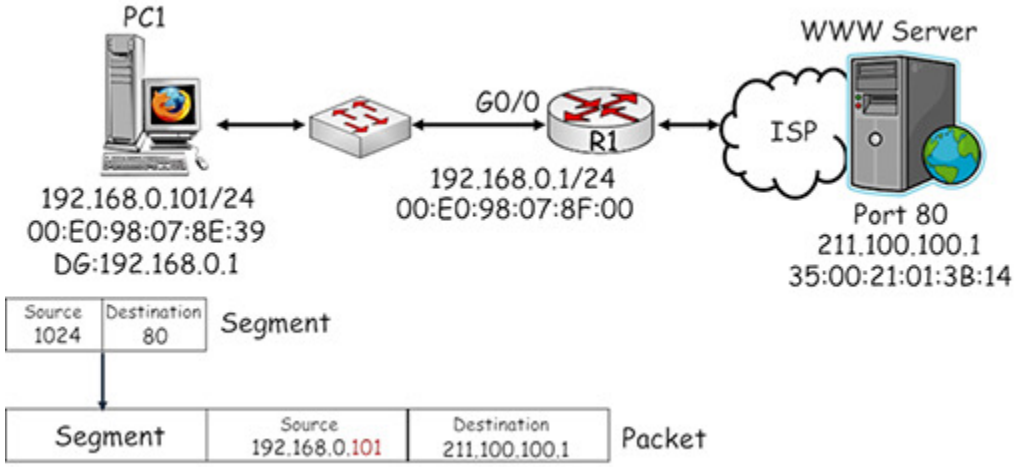
PC1 creates a local table called an ARP cache (table) in which it stores all the IP address/MAC address pairings it has learnt.
As a result, subsequent frames created by PC1 addressed to the printer will use the ARP cache to find the required destination address instead of using ARP.

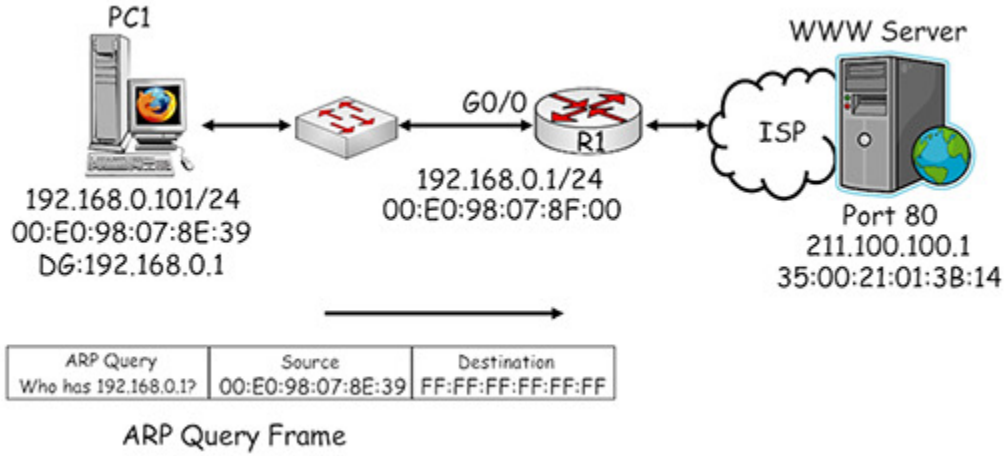
The entries in the ARP cache have a lifetime associated with them, which is constantly updated while the device is sending frames using the entries.

Once the device stops sending frames the entries will timeout and be removed from the cache. This prevents the cache becoming full of outdated MAC address information.

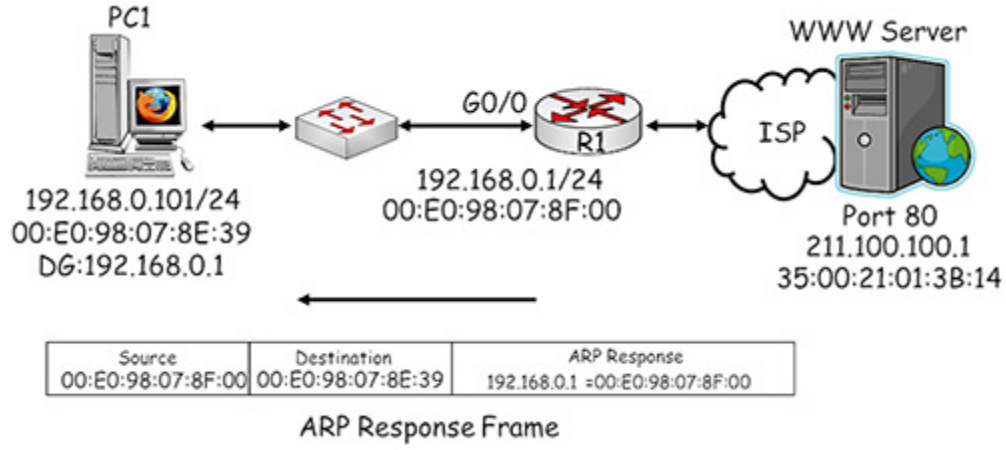


Sending data to the WAN

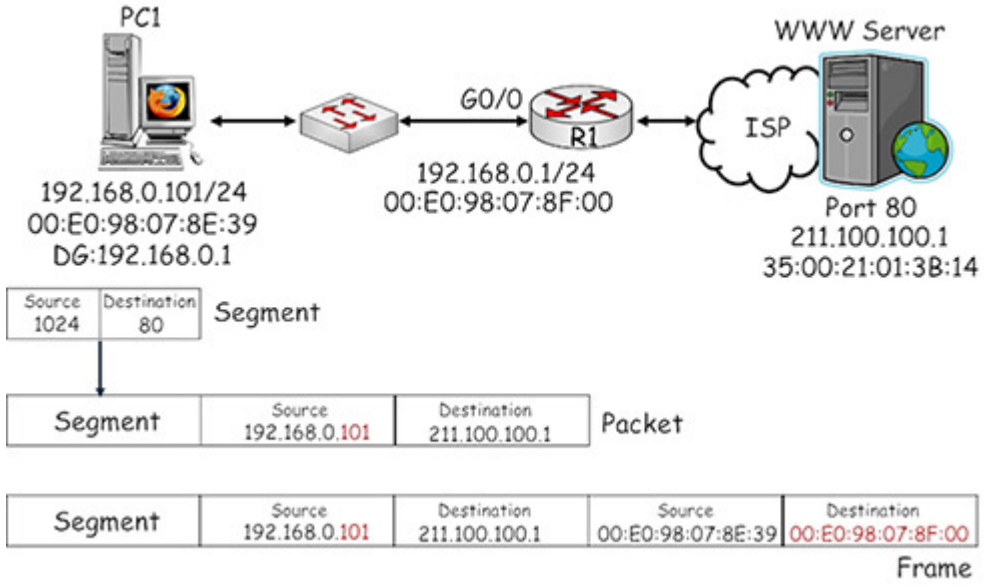


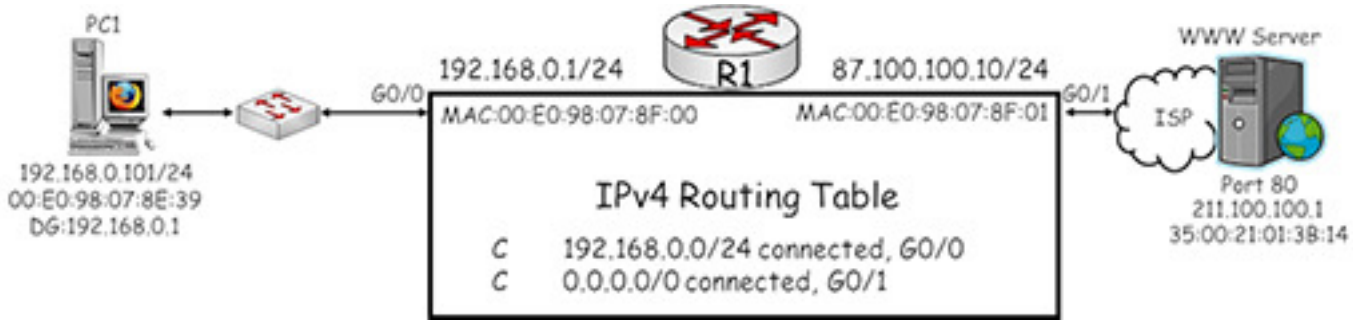


PC1 recognises that the destination address is on a different IP network and that it must send packets via the default gateway it has been configured to use: 192.168.0.1. PC1 encapsulates the webpage request into a succession of TCP segments, which are then encapsulated in appropriately addressed IP packets. The source address identifies PC1 (192.168.0.101) and the destination address identifies the web server (211.100.100.1):



Sending data to the WAN





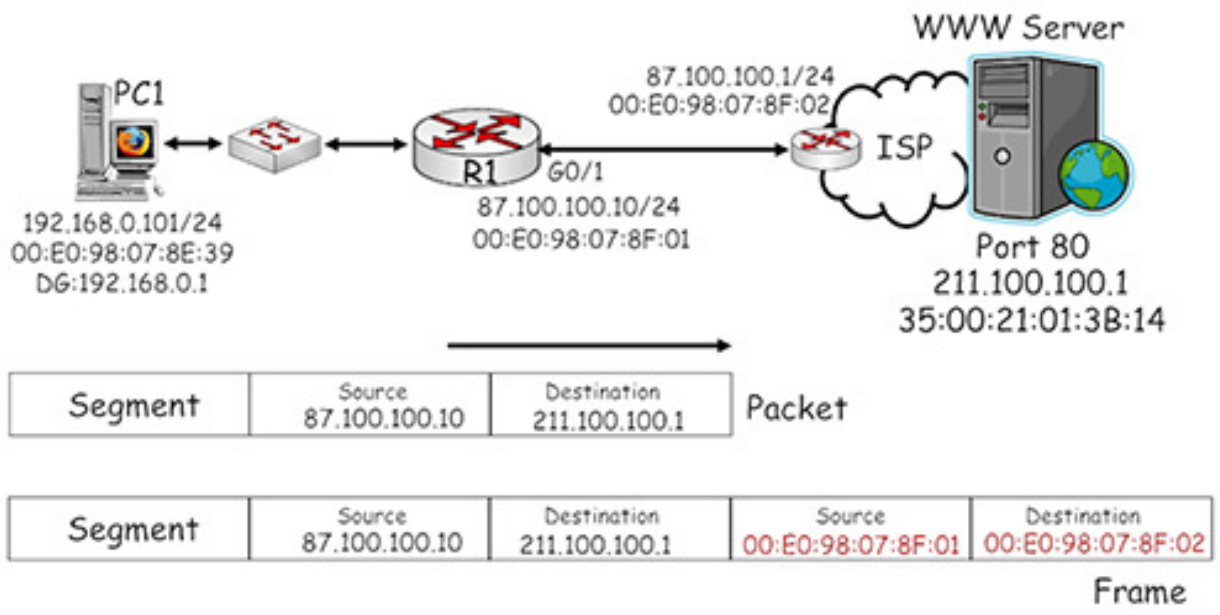
PC1 uses the MAC address it received in the ARP response to complete the destination MAC address field in the frame it is using to send data to the default gateway.

Check what Destination IP and MAC addresses are!

Dest IP of WWW Server and Dest MAC of R1 G0/0

The frame is delivered across the local network to the gigabit interface of the home router, R1. Because the destination MAC address of the frame matches the MAC assigned to the interface, the router accepts the frame and de-encapsulates it to recover the packet. The router then tries to match the destination IP address with an entry within its own routing table so it can make a forwarding decision:

Sending data to the WAN



The G0/1 interface is connected via whichever broadband technology is being used (DSL, cable or wireless) to a router within the ISP, which is configured with IP and MAC addresses. Note that the source IP address of the packet has been converted by NAT to 87.100.100.10, which is the public IP address that uniquely identifies the home router within the Internet.

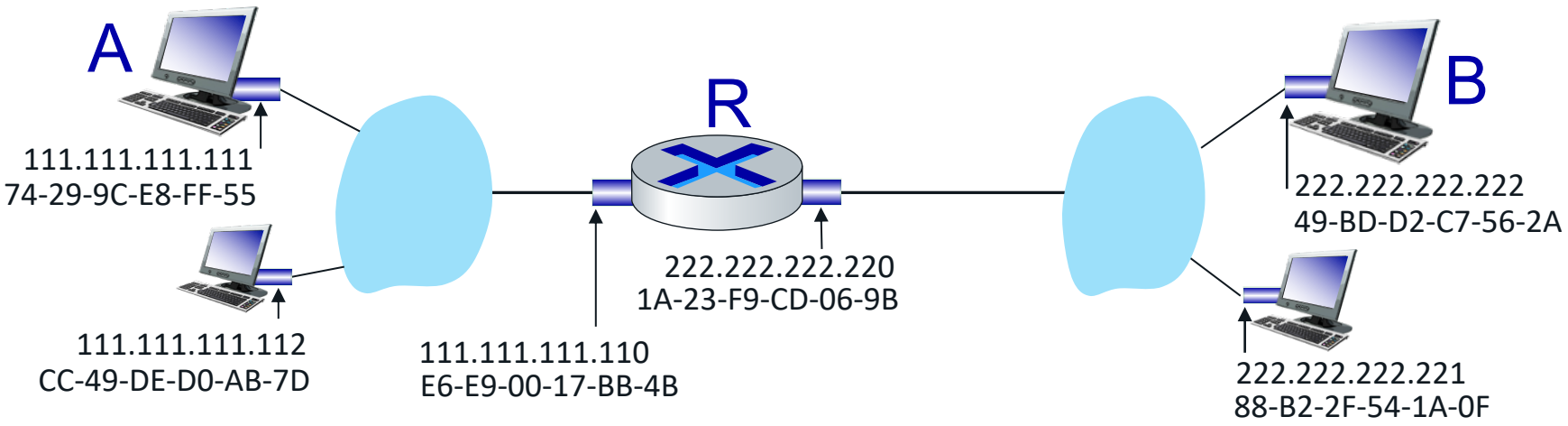
The packet is then encapsulated within an Ethernet frame, which uses the MAC address of home router interface G0/1 as its source and the MAC address of the ISP router interface as its destination.

Subsequent routers that forward the packet towards the WWW server and will not change the source IP address, otherwise reply packets would not be able to locate the home router. The packet will be encapsulated in a new frame every time it is forwarded by a router. The frames that are used may not be Ethernet – it depends on the type of WAN technology that is utilised by the devices which forward the packet to its destination. Another function provided by routers is to limit the spread of broadcast traffic such as ARP. Imagine what would happen if ARP could be propagated across the Internet – every time an ARP was generated, on any device, it would be sent to every other device in the world. This is obviously extremely undesirable and router interfaces create a broadcast domain – they will examine broadcast traffic, but they will not forward it onto other networks.

Routing to Another Subnet

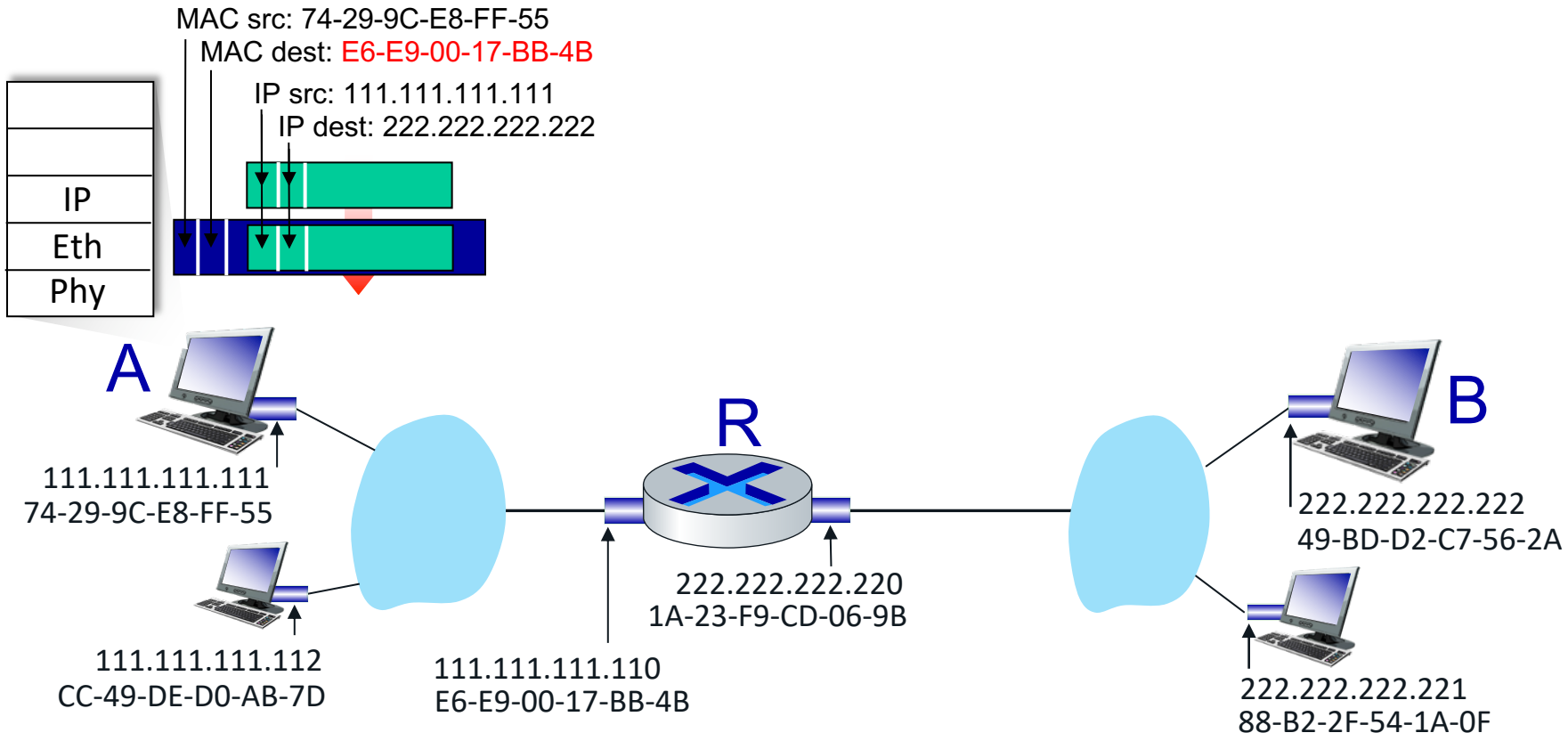
walkthrough: sending a datagram from *A* to *B* via *R*

- focus on addressing – at IP (datagram) and MAC layer (frame) levels
- assume that:
  - A knows B's IP address
  - A knows IP address of first hop router, R (how?)
  - A knows R's MAC address (how?)



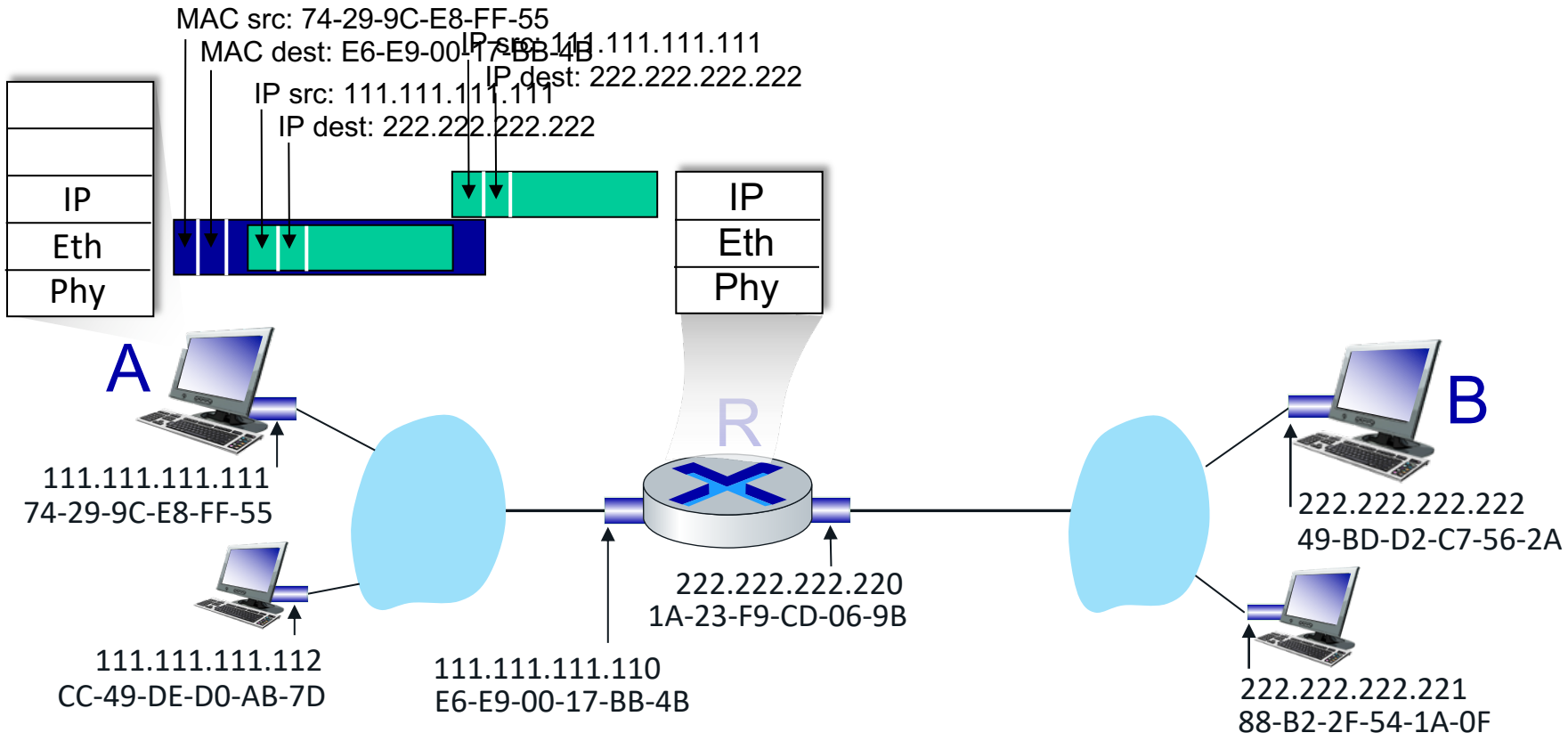
Routing to Another Subnet

- A creates IP datagram with IP source A, destination B
- A creates link-layer frame containing A-to-B IP datagram
  - R's MAC address is frame's destination



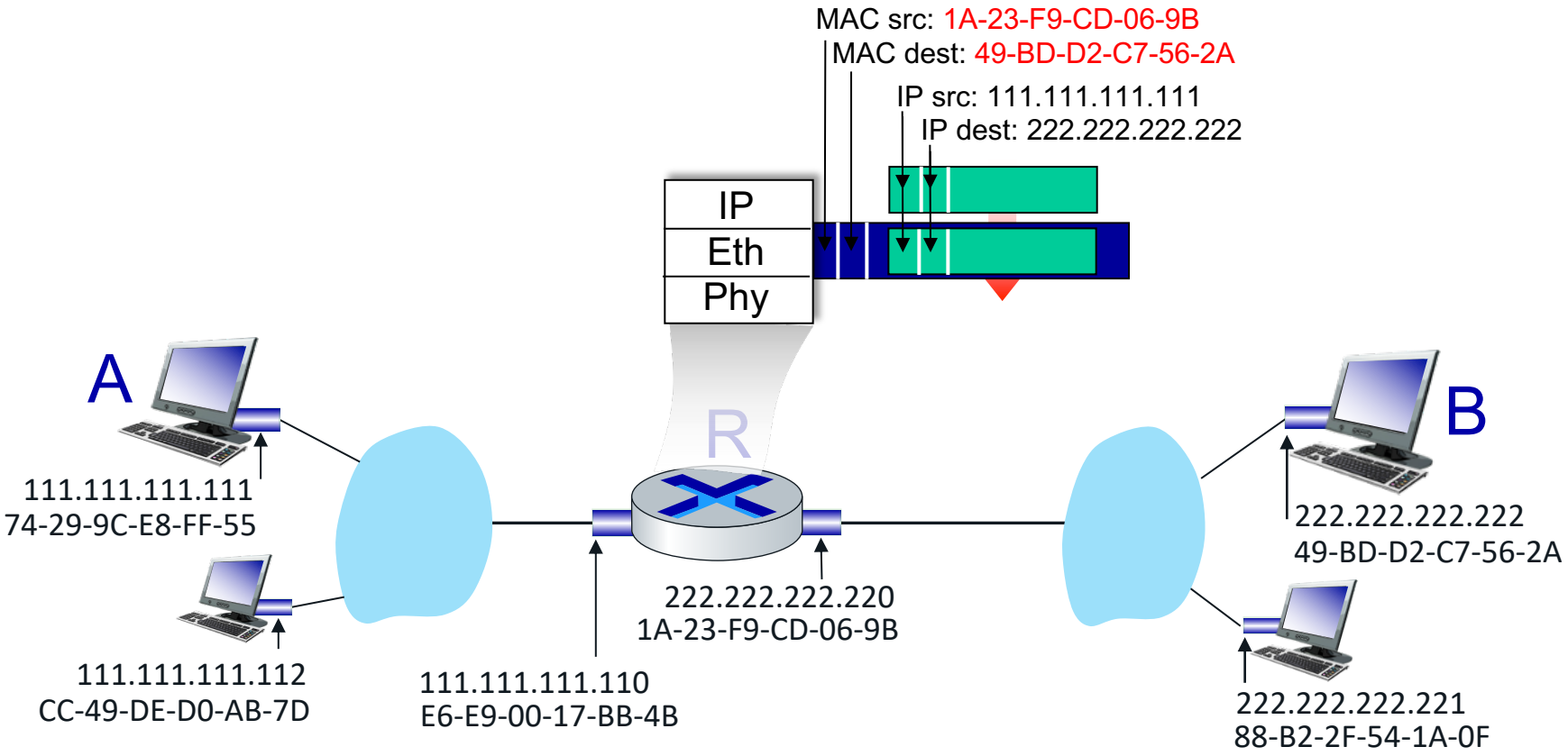
Routing to Another Subnet

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



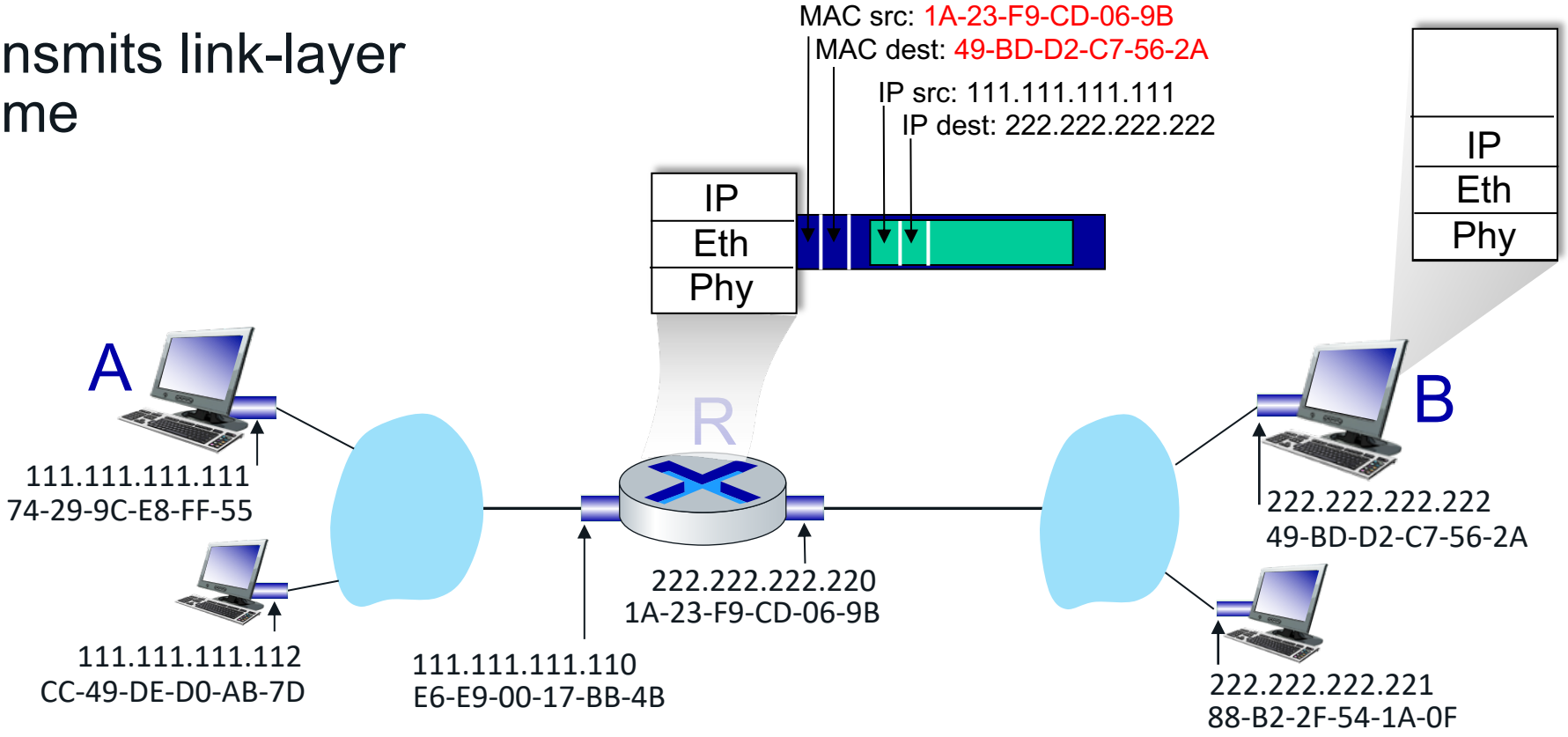
Routing to Another Subnet

- R determines outgoing interface, passes datagram with IP source A, destination B to link layer
- R creates link-layer frame containing A-to-B IP datagram. Frame destination address: B's MAC address



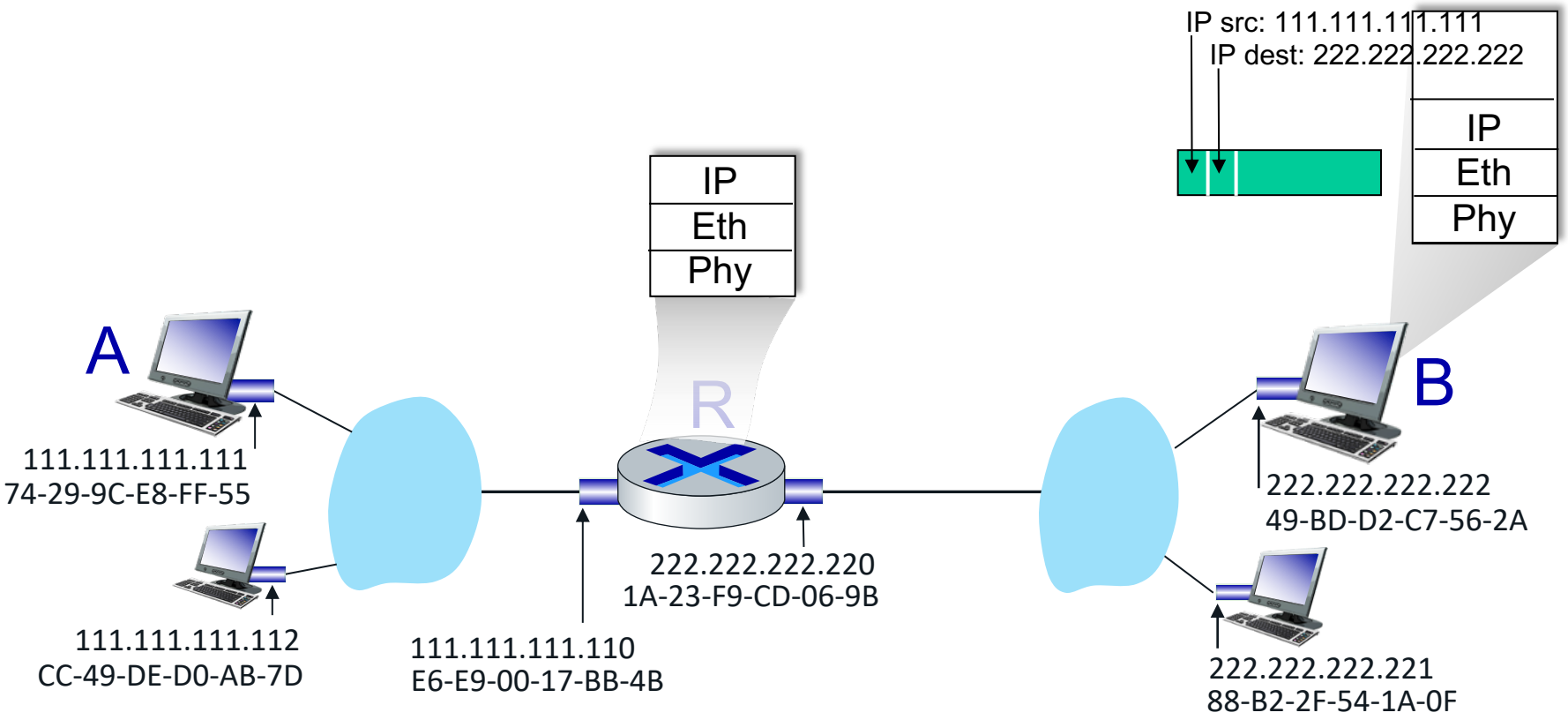
Routing to Another Subnet

- R determines outgoing interface, passes datagram with IP source A, destination B to link layer
- R creates link-layer frame containing A-to-B IP datagram. Frame destination address: B's MAC address
- transmits link-layer frame

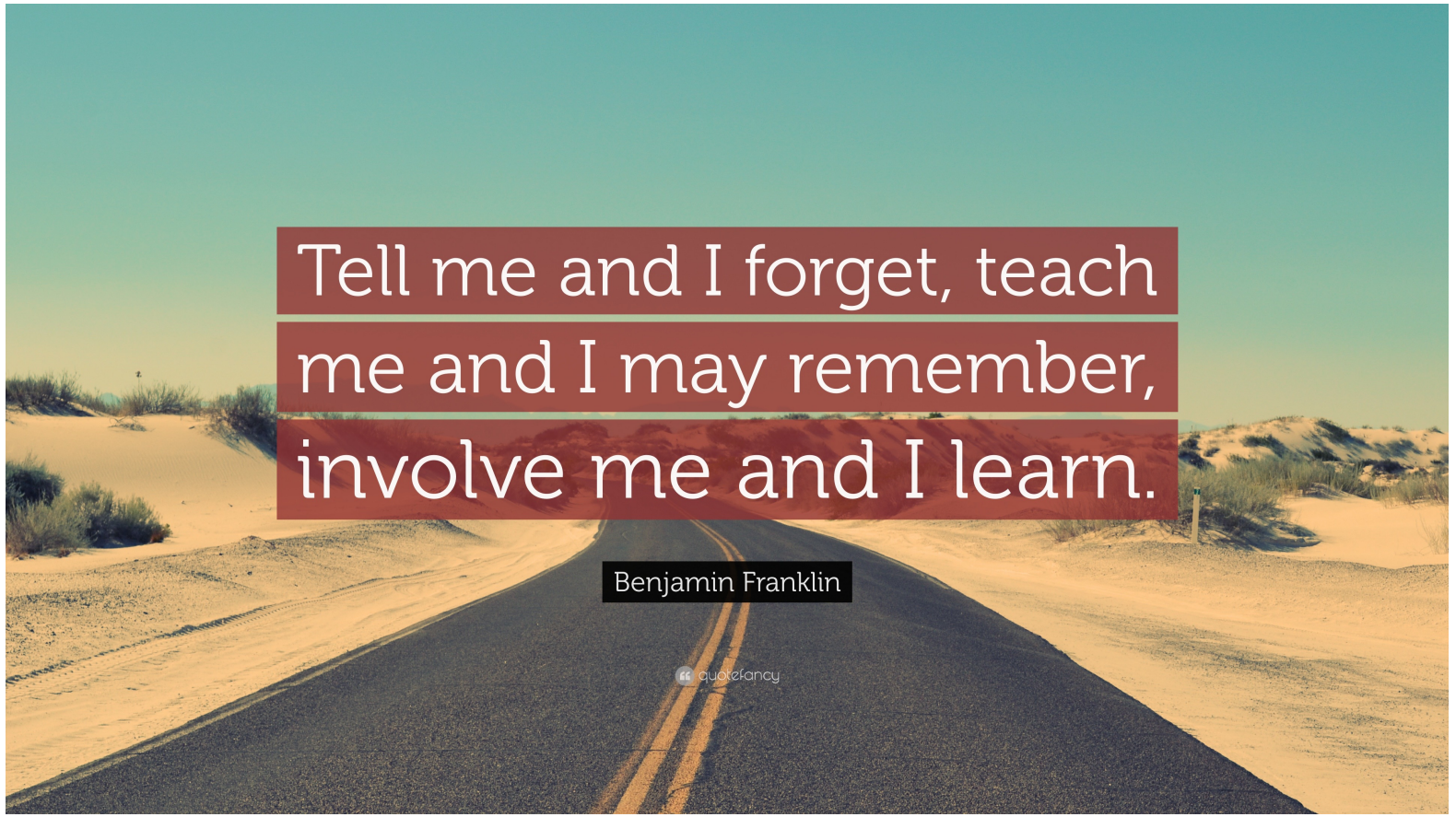


Routing to Another Subnet

- B receives frame, extracts IP datagram destination B
- B passes datagram up protocol stack to IP



Thank You!


Tell me and I forget, teach
me and I may remember,
involve me and I learn.
Benjamin Franklin
quotefancy

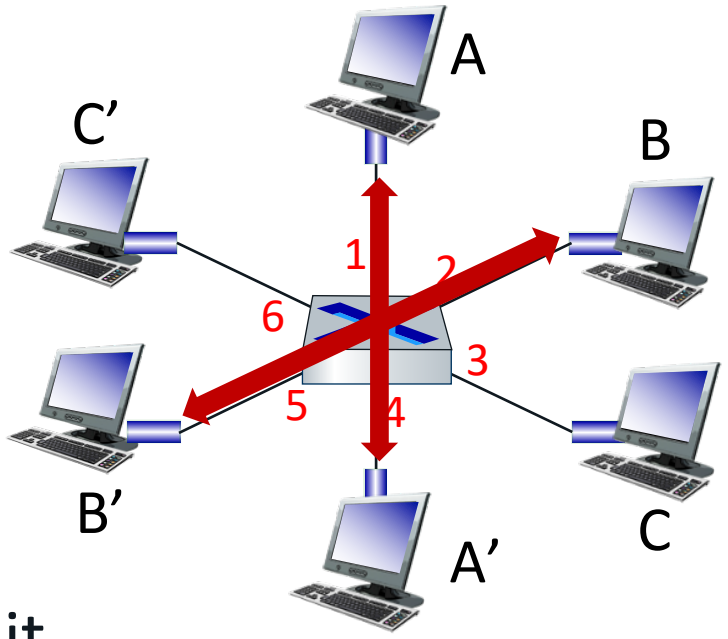
Switch and Router

Ethernet Switch

- Switch is a link-layer device: takes an *active* role
  - store, forward Ethernet frames
  - examine incoming frame's MAC address, *selectively* forward frame to one-or-more outgoing links when frame is to be forwarded on segment, uses CSMA/CD to access segment
- transparent: hosts *unaware* of presence of switches
- plug-and-play, self-learning
  - switches do not need to be configured

Switch: Multiple Simultaneous Transmissions

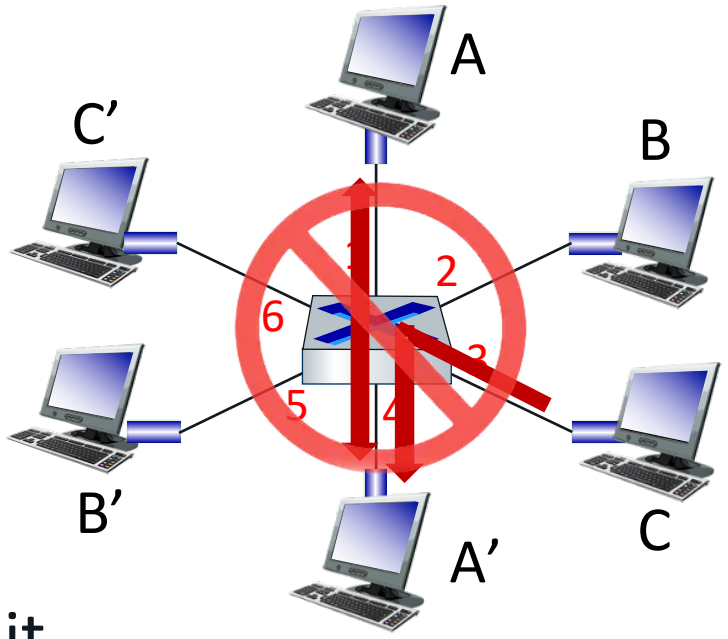
- hosts have dedicated, direct connection to switch
- switches buffer packets
- Ethernet protocol used on *each* incoming link, so:
  - no collisions; full duplex
  - each link is its own collision domain
- switching: A-to-A' and B-to-B' can transmit simultaneously, without collisions



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

Switch: Multiple Simultaneous Transmissions

- hosts have dedicated, direct connection to switch
- switches buffer packets
- Ethernet protocol used on *each* incoming link, so:
  - no collisions; full duplex
  - each link is its own collision domain
- switching: A-to-A' and B-to-B' can transmit simultaneously, without collisions
  - but A-to-A' and C to A' can *not* happen simultaneously



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

Switch: Forwarding Table

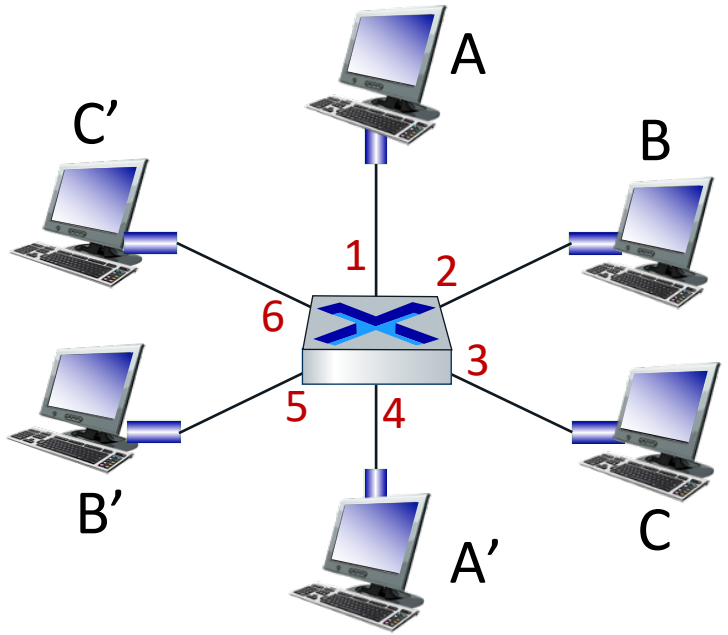
Q: how does switch know A' reachable via interface 4, B' reachable via interface 5?

A: each switch has a switch table, each entry:

- (MAC address of host, interface to reach host, time stamp)
- looks like a routing table!

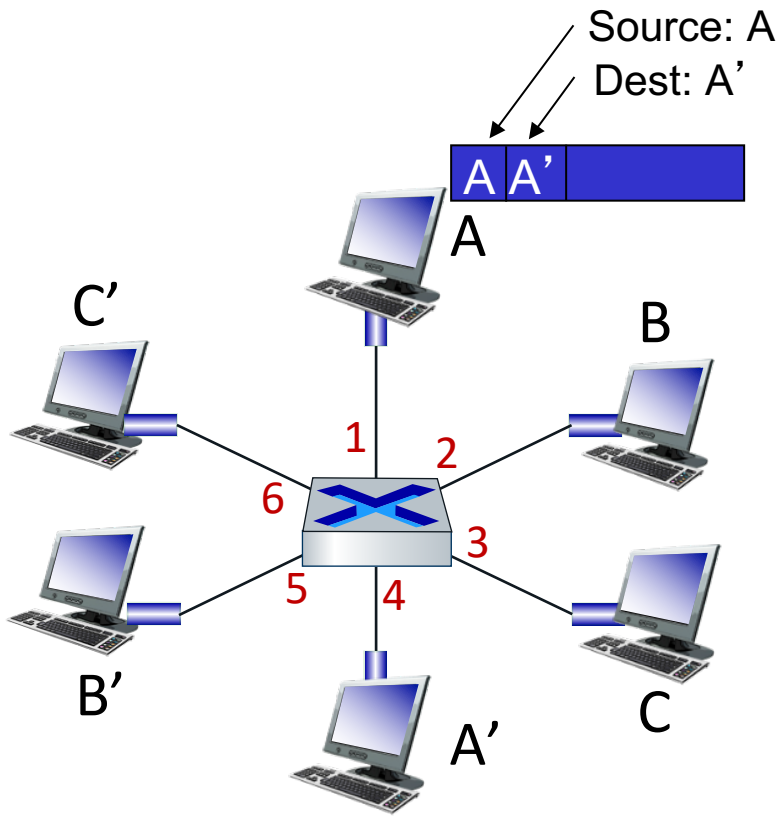
Q: how are entries created, maintained in switch table?

- something like a routing protocol?



Switch: Self-Learning

- 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

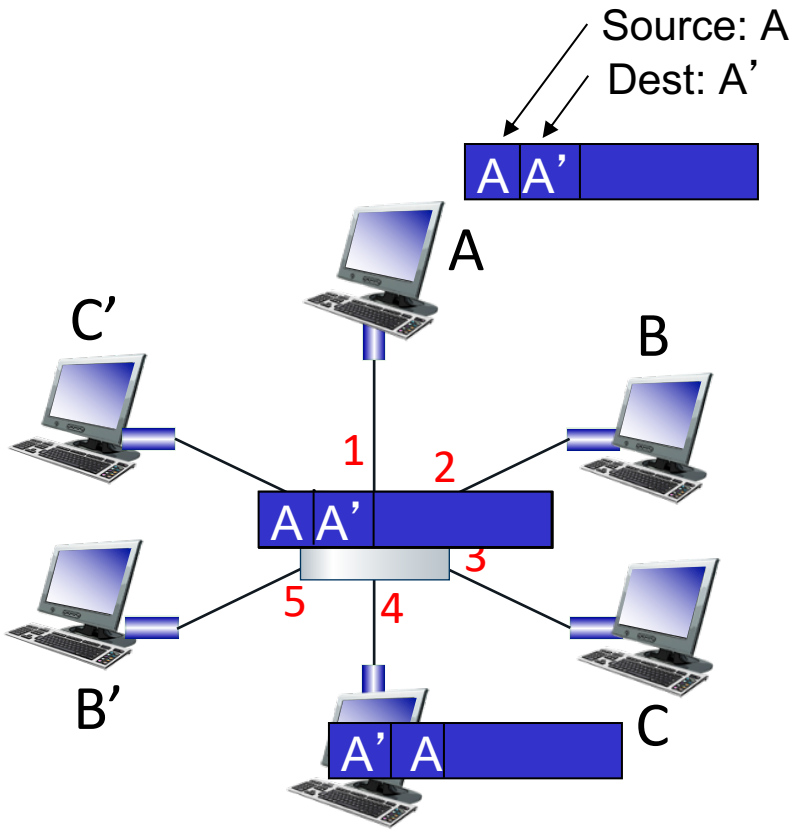


MAC addr	interface	TTL
A	*1*	*60*

Switch table (initially empty)

Switch Example: Self-Learning and Forwarding

- frame destination, A', location unknown: flood
- destination A location known: selectively send on just one link



MAC addr	interface	TTL
A	*1*	*60*
A'	*4*	*60*

switch table (initially empty)

What is a router?

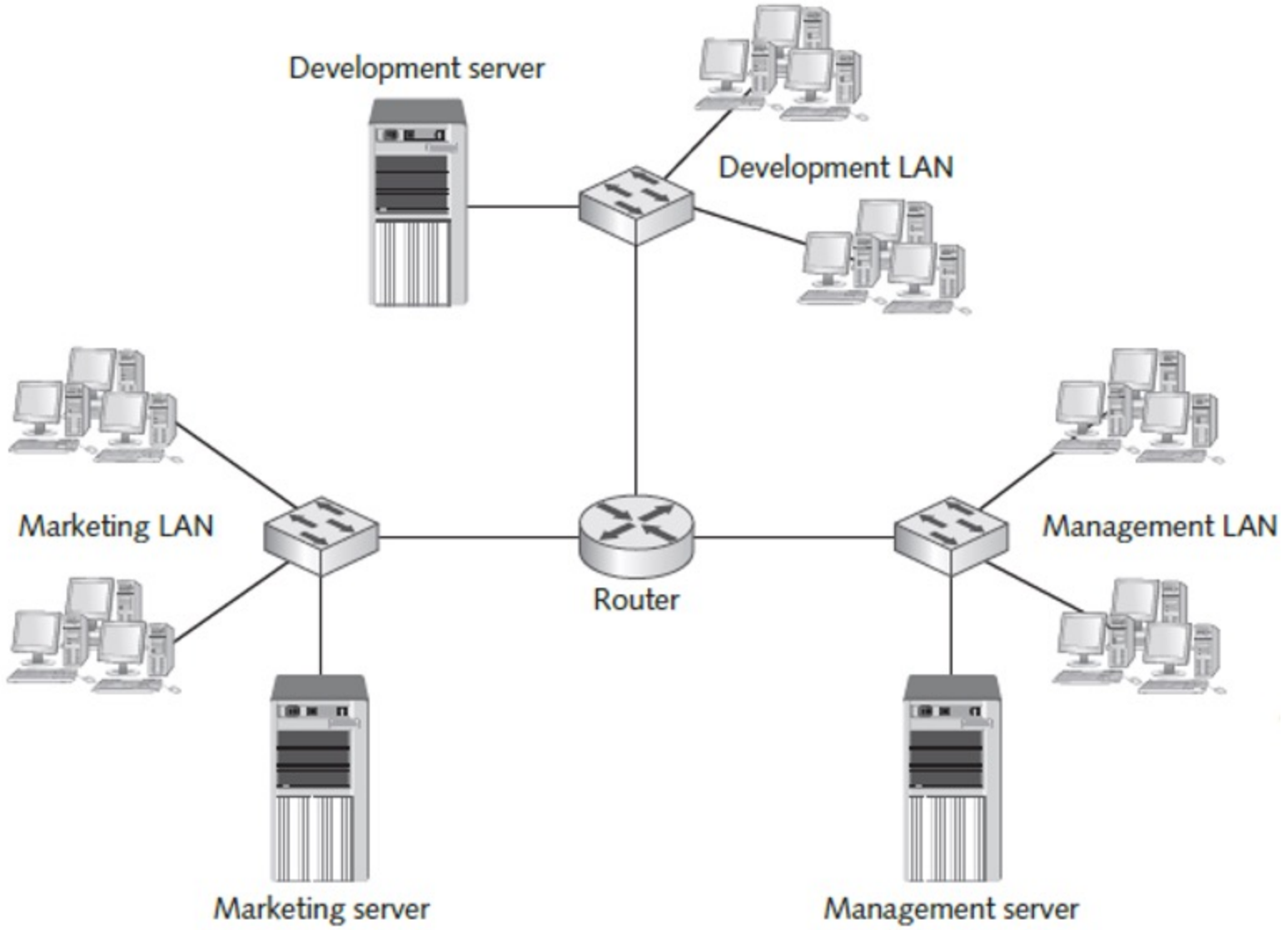
A router is a networking device that connects computer networks, for example, connecting a home network with the Internet.

Routers are devices that enable multiple LANs to communicate with one another by forwarding packets from one LAN to another

Routers operate at Layer 3 (network layer); a router uses the destination IP address in a data packet to determine where to forward the packet.

Eg. 3 Tbps 7950 XRS

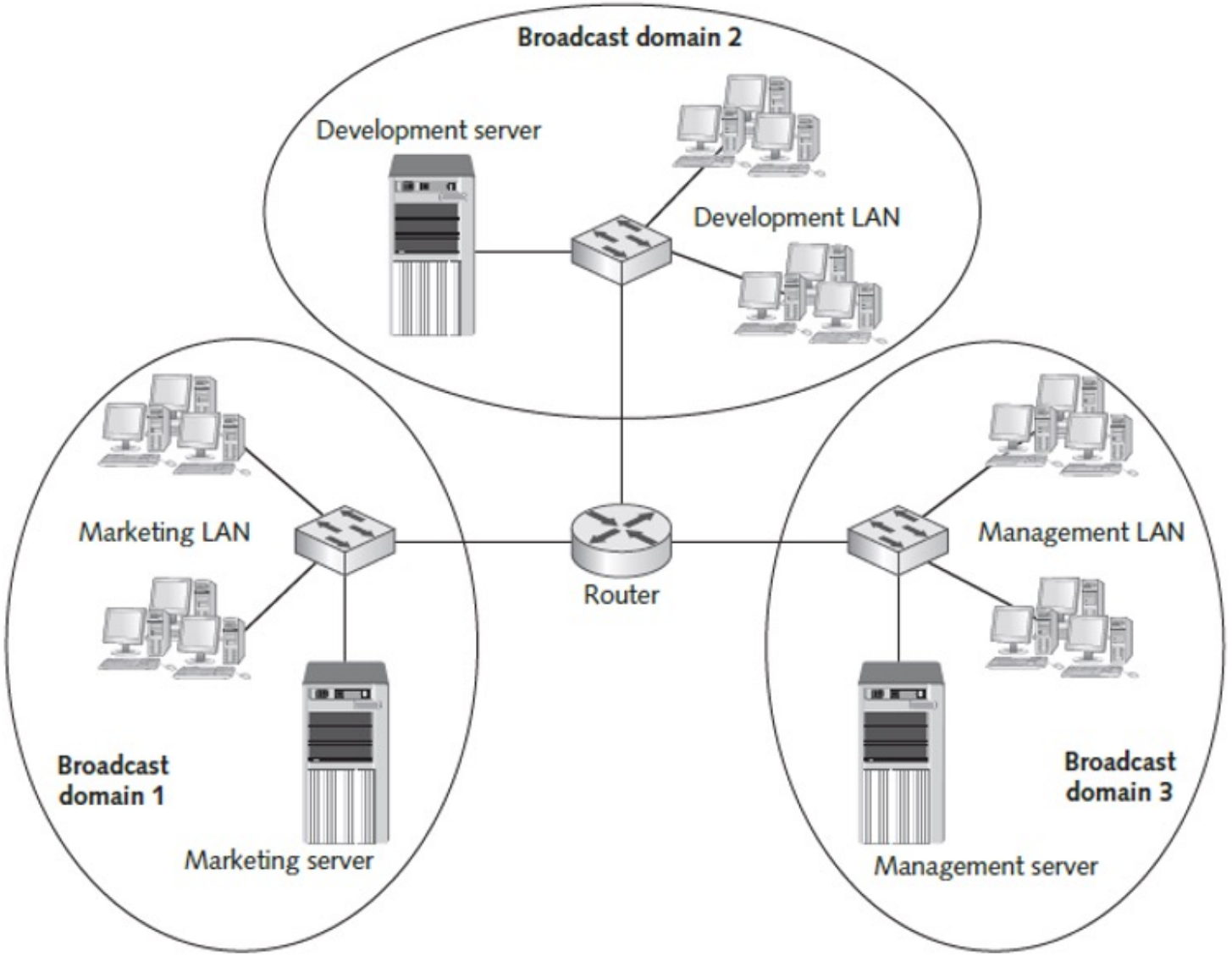
What is a router? – It connects LANs



What is a router? – It creates broadcast domains

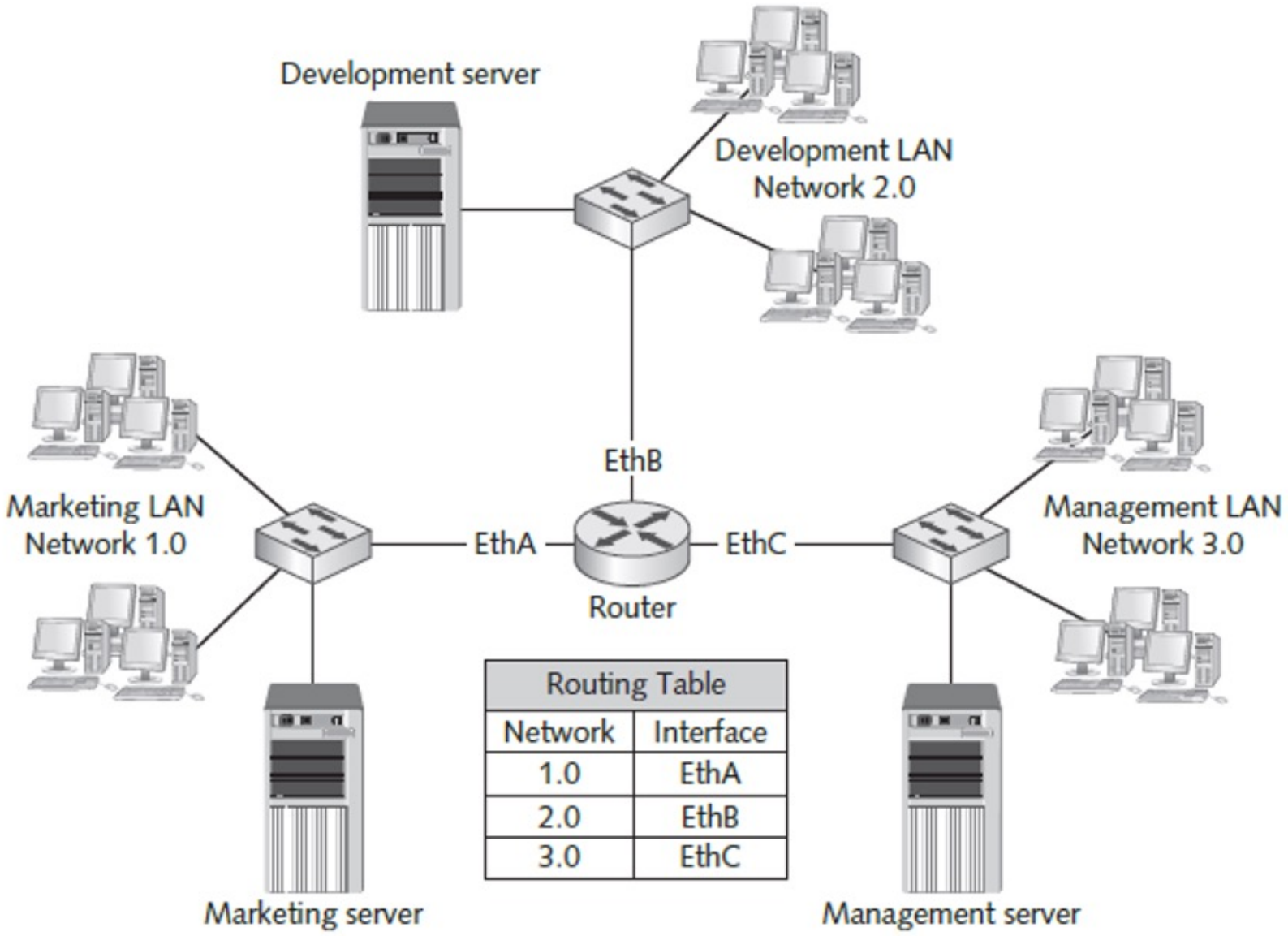
The scope of devices to which broadcast frames are forwarded is called a **broadcast domain**

Each router interface in a network creates another broadcast domain



What is a router? – It creates broadcast domains

Routers maintain routing tables composed of IP network addresses and interface pairs to determine where to forward packets on an internetwork



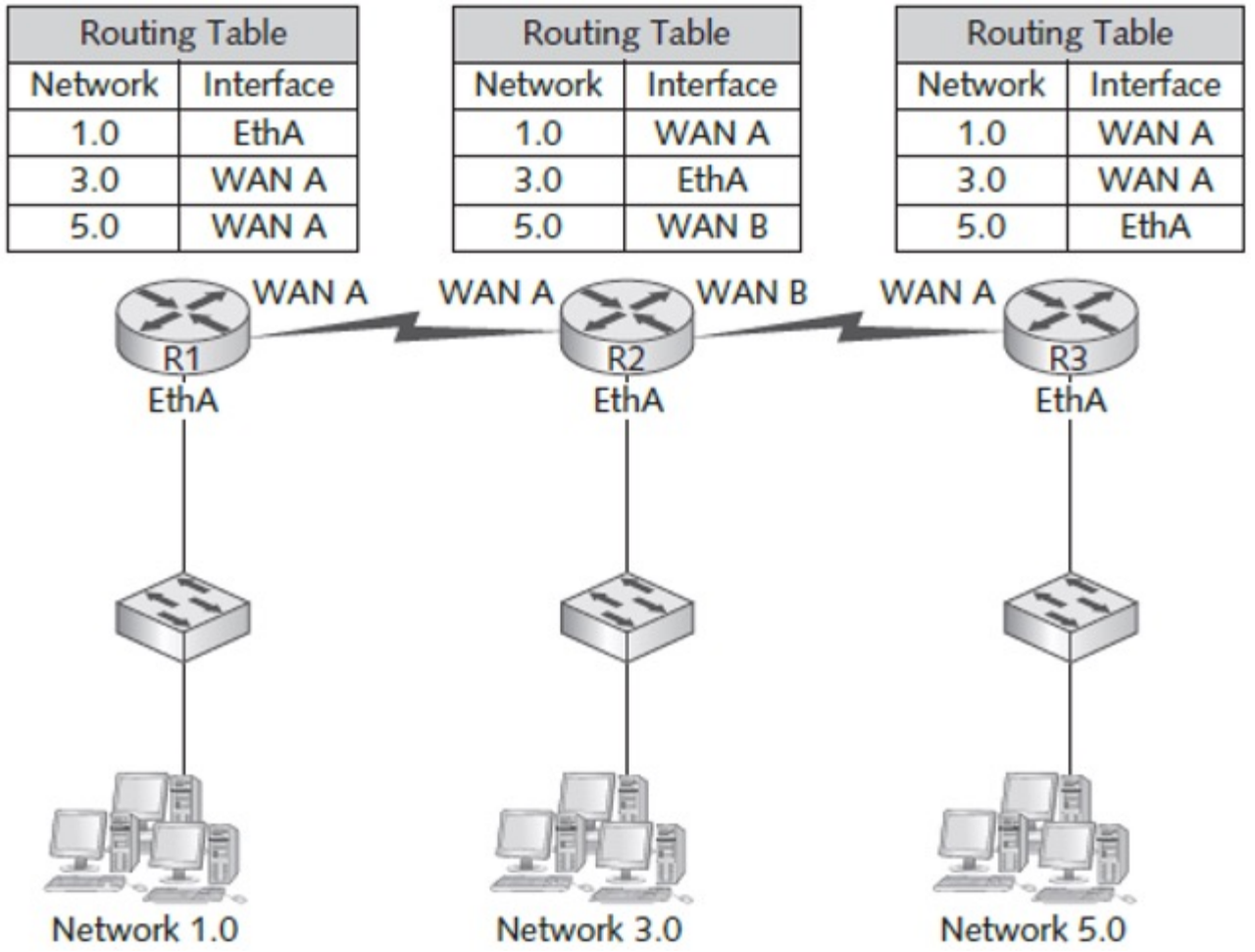
What is a router? – It uses route table

Routers maintain routing tables composed of

- IP network addresses and
- interfaces

to determine where to forward packets on an internetwork

>netstat -r



Route Table and ARP Table

shihaoy — netstat -r — 80×24

```
For more details, please visit https://support.apple.com/kb/HT208050.
C02FW0B8MD6N:~ shihaoy$ netstat -r
Routing tables

Internet:
Destination        Gateway            Flags        Netif Expire
default            10.23.65.1         UGScg          en7
10.23.65/24        link#12            UCS            en7      !
10.23.65.1/32      link#12            UCS            en7      !
10.23.65.1         0:0:5e:0:1:48      UHLWIir        en7   1199
10.23.65.33        14:3f:a6:21:e9:b7  UHLWIi         en7    917
10.23.65.115/32    link#12            UCS            en7      !
127                localhost          UCS            lo0
localhost          localhost          UH             lo0
169.254            link#12            UCS            en7      !
224.0.0/4          link#12            UmCS           en7      !
224.0.0.251        1:0:5e:0:0:fb      UHmLWI         en7
239.255.255.250    1:0:5e:7f:ff:fa    UHmLWI         en
255.255.255.255/32 link#12            UCS            en
```

shihaoy — -bash — 80×24

```
Last login: Wed Feb 23 13:45:58 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
C02FW0B8MD6N:~ shihaoy$ arp -a
? (10.23.65.1) at 0:0:5e:0:1:48 on en7 ifscope [ethernet]
? (10.23.65.33) at 14:3f:a6:21:e9:b7 on en7 ifscope [ethernet]
? (224.0.0.251) at 1:0:5e:0:0:fb on en7 ifscope permanent [ethernet]
? (239.255.255.250) at 1:0:5e:7f:ff:fa on en7 ifscope permanent [ethernet]
C02FW0B8MD6N:~ shihaoy$
```

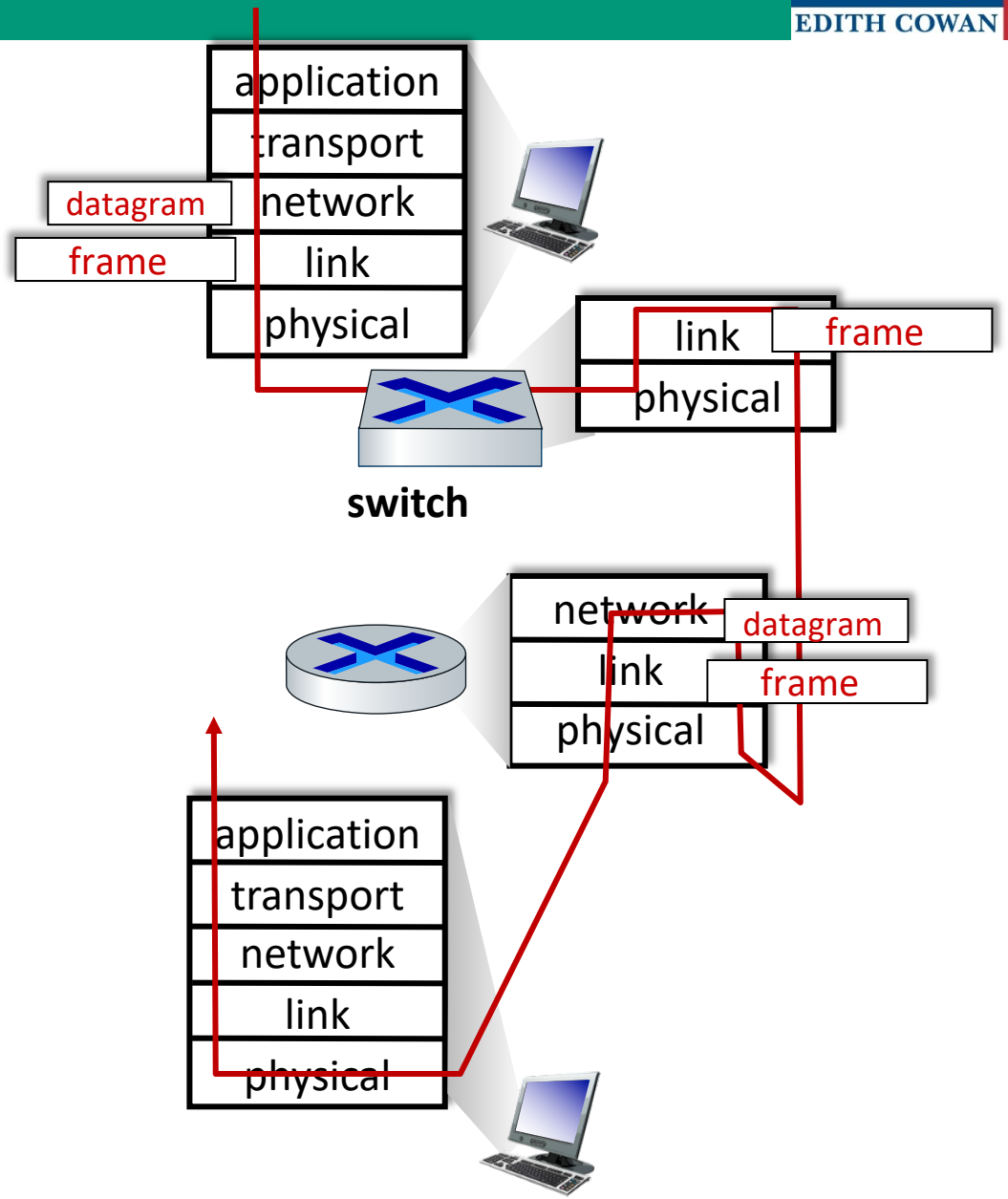
Switch and Router

both are store-and-forward:

- *routers*: network-layer devices (examine network-layer headers)
- *switches:* link-layer devices (examine link-layer headers)

both have forwarding tables:

- *routers:* compute tables using routing algorithms, IP addresses
- *switches:* learn forwarding table using flooding, learning, MAC addresses



Thank You!


Tell me and I
forget. Teach me
and I remember.
Involve me and I
learn.
~ Benjamin Franklin