Chapter 5
Link Layer

Teacher : Xu Yang

Chapter 5: Link layer

our goals:

- understand principles behind link layer services:
  - error detection, correction
  - sharing a broadcast channel: multiple access
  - link layer addressing
  - local area networks: Ethernet
- instantiation, implementation of various link layer technologies

Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

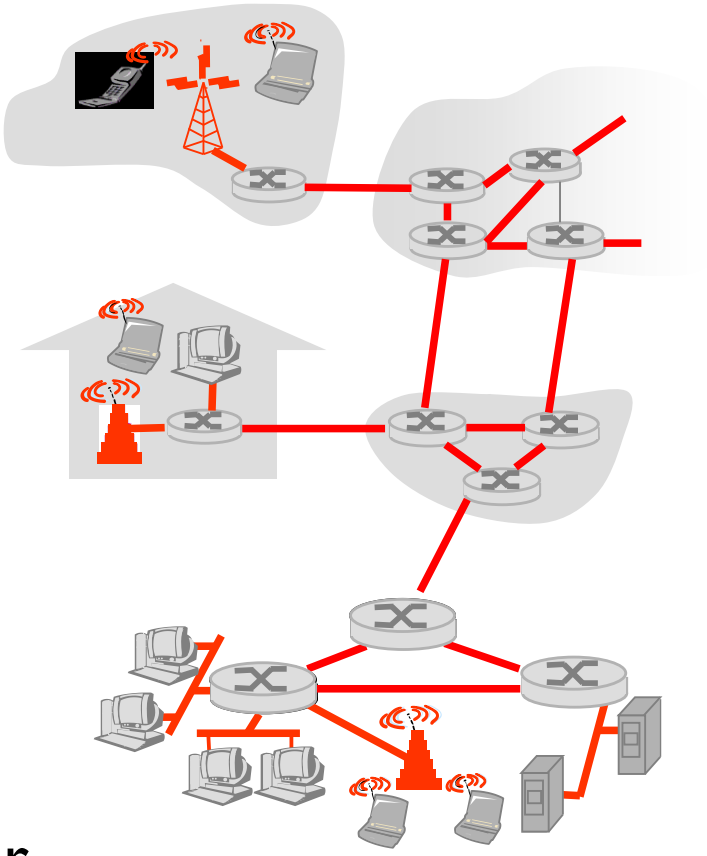
5.3 multiple access protocols

5.4 LANs

- addressing, ARP
- Ethernet
- switches

The Link Layer

- Network layer provides logical communication between two hosts.



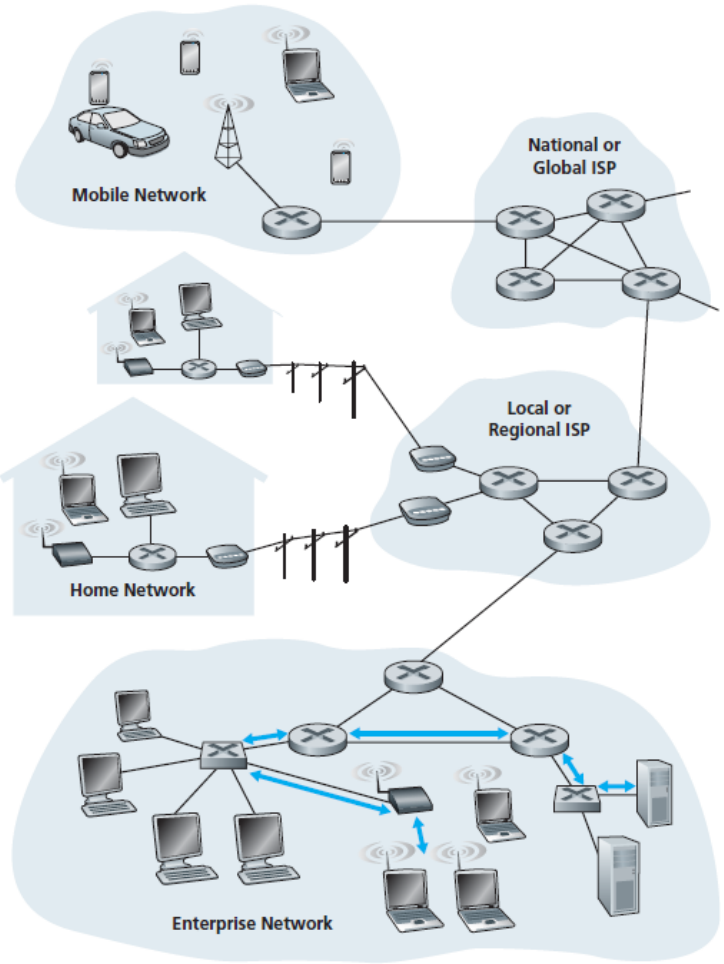
- The communication path between two hosts consists of multiple individual links.
- These individual links may be different
  - Point-to-point link
  - Broadcast link

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

- Different link layer protocols are used for different types of links.
  - e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link

Link layer: introduction

- Any devices that runs a link layer protocol (hosts and routers) as a node
- communication channels that connect adjacent nodes along communication path: links
  - wired links
  - wireless links
  - LANs
- frame, : Link-layer packet is called a **frame.** It is the unit of data transmitted by a link-layer protocol. Link layer protocols encapsulate each network-layer datagram into a link-layer frame before transmission over a link.



Link layer services

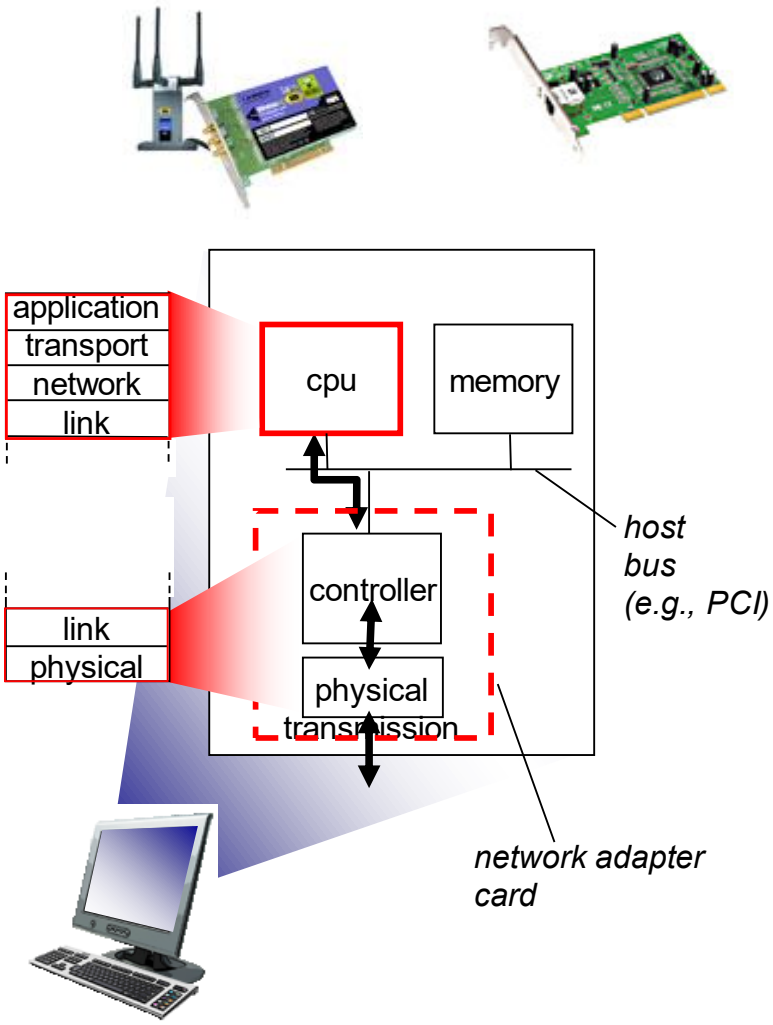
- *Link layer protocol is to move a datagram from one node to adjacent node over a link. Each link layer protocol provides different services in terms of different aspects*
- *Framing:* A frame consists of a data field, in which the network-layer datagram is inserted, and a number of header fields. The structure of the frame is specified by the link-layer protocol.
  - "MAC" addresses used in frame headers to identify source, dest
- *reliable delivery between adjacent nodes :* It aims to guarantee to transfer each network-layer datagram over each link without error.
  - can be achieved with acknowledgments and retransmissions
  - The reliable delivery is achieved through error detection and correction.
    - Error Detection: receiving node detects the presence of errors. Then it signals the sender for retransmission or just drops the corrupted frame.
    - Error Correction: mechanism for the receiving node to locate and correct the error.
  - seldom used on low bit-error link (fiber, some twisted pair)
  - wireless links: high error rates

Link layer services (more)

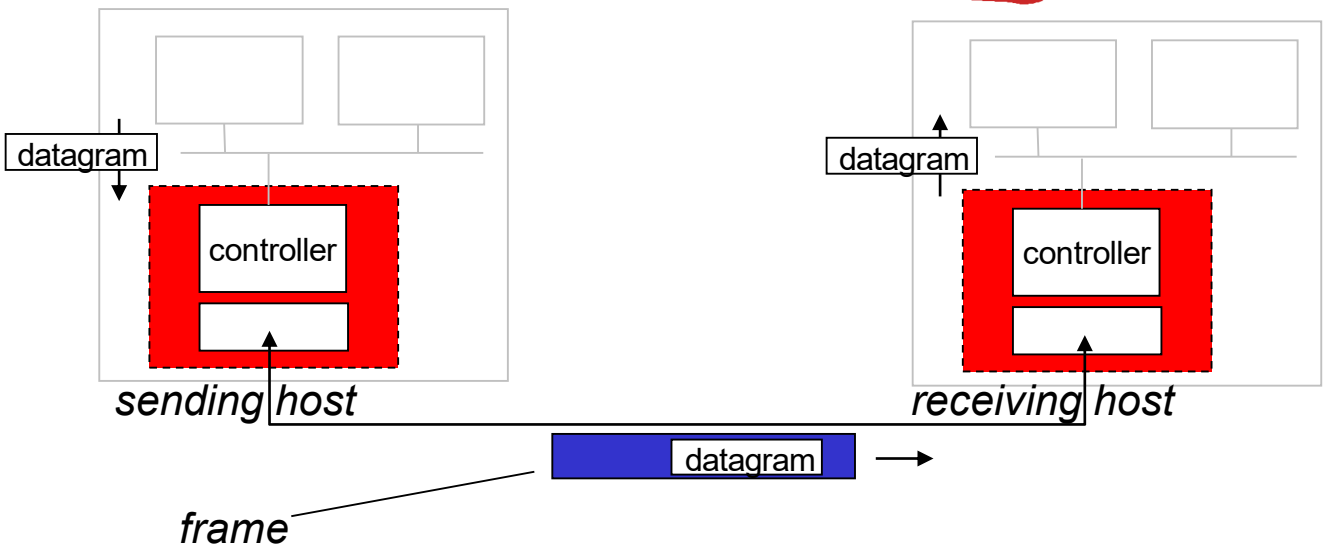
- *Link access:* *A medium access control (MAC) protocol specifies the rules by which a frame is transmitted over the link.*
  - For point-to-point links where have a single sender at one end of the link and a single receiver at the other end of the link, the MAC protocol is simple. The sender can send frames whenever the link is idle.
  - For shared links where multiple nodes share a single broadcast link, the MAC protocol coordinates the frame transmissions of multiple nodes.
- *Flow control:* *It is a kind of rate matching between sender and receiver. The object is to prevent the transmitting node from transmitting packets at a rate higher than the speed that the receiving node can process.*
- *half-duplex and full-duplex*
  - with half duplex, nodes at both ends of link can transmit, but not at same time

Where is the link layer implemented?

- in each and every host
- link layer implemented in "adaptor" (aka *network interface card* NIC) or on a chip
  - Ethernet card, 802.11 card; Ethernet chipset
  - implements link, physical layer
- attaches into host's system buses
- combination of hardware, software, firmware



Adaptors communicating



- sending side:
  - encapsulates datagram in frame
  - adds error checking bits, rdt, flow control, etc.
  - Implements channel access for shared medium, and transmits on link
- receiving side
  - looks for errors, rdt, flow control, etc
  - extracts datagram, passes to upper layer at receiving side

Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

5.3 multiple access protocols

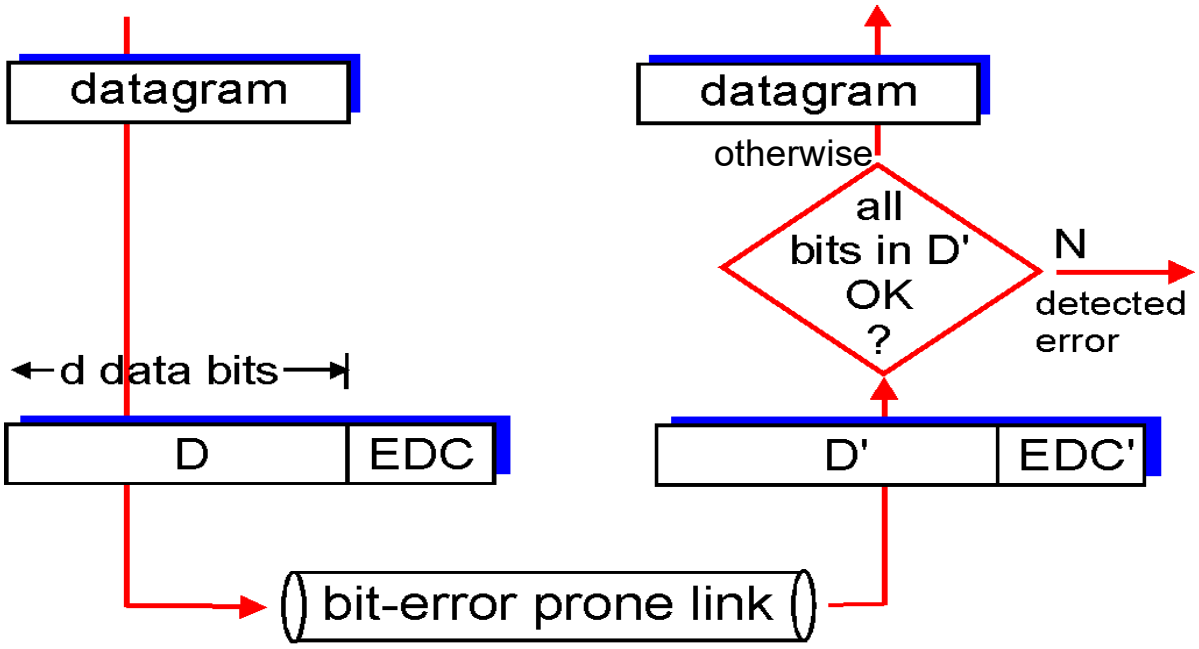
5.4 LANs

- addressing, ARP
- Ethernet
- switches

Error detection

EDC= Error Detection and Correction bits (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



Three error detection techniques

Three error detection techniques

- Parity checking
- Checksuming
- Cyclic Redundancy Check (CRC)

Parity checking

single bit parity:

- *d*etect single bit errors

two-dimensional bit parity:

- detect and correct single bit errors

Two Schemes:

- Even parity scheme
- Odd parity scheme

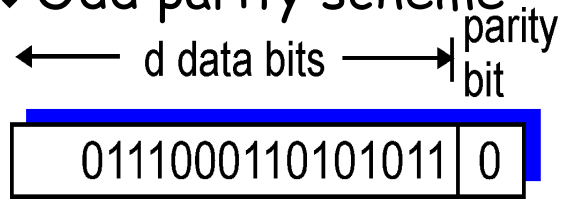
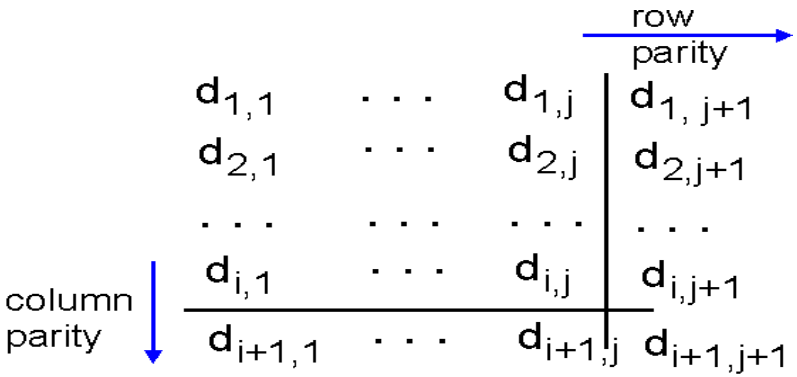
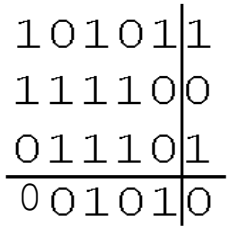


Figure. One-bit odd parity

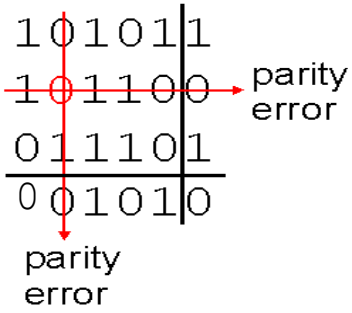


In an odd parity scheme, the sender simply includes one additional bit and chooses its value such that the total number of 1s in the d + 1 bits (the original information plus a parity bit) is odd.

The receiver need only count the number of 1s in the received d + 1 bits.



no errors



correctable single bit error

Internet checksum (review)

goal: detect "errors" (e.g., flipped bits) in transmitted packet (note: used at transport layer *only*)

sender:

- treat segment contents as sequence of 16-bit integers
- checksum: addition (1's complement sum) of segment contents
- sender puts checksum value into UDP checksum field

receiver:

- compute checksum of received segment
- check if computed checksum equals checksum field value:
  - NO - error detected
  - YES - no error detected. *But maybe errors nonetheless?*

Internet checksum: example

1's complement sum

example: add two 16-bit integers

	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
wraparound	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1
sum		1	0	1	1	1	0	1	1	1	0	1	1	1	1	0	0
checksum		0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	1

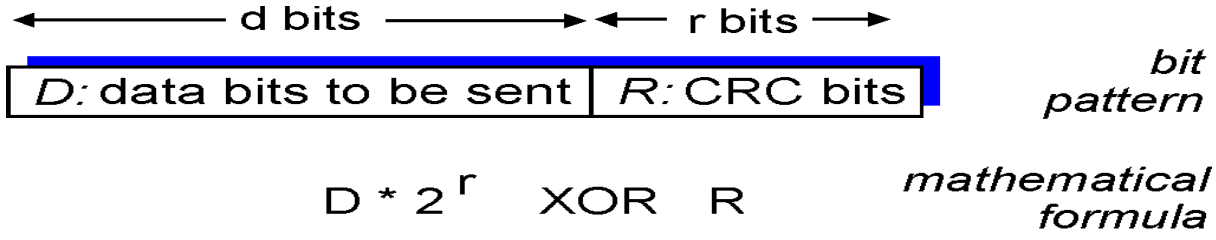
1. Add the 16-bit values up. Each time a carry-out (17th bit) is produced, swing that bit around and add it back into the LSb (one's digit). This is somewhat erroneously referred to as "one's complement addition."
2. Once all the values are added in this manner, invert all the bits in the result. A binary value that has all the bits of another binary value inverted is called its "one's complement," or simply its "complement."

Cyclic redundancy check (CRC)

- more powerful error-detection coding technique widely used in computer networks (Ethernet, 802.11 WiFi, ATM).
- A message is sent with a calculated CRC code such that it may be verified by receiver

Cyclic Redundancy Check (CRC)

- Basic idea of CRC: For a given data D with d bits,
  - choose r+1 bit pattern (known as a generator), G, have known by both sender and receiver.
  - Sender: It calculates additional CRC code with r bits, say R, and append them to D before transmission such that the resulting (d+r) bit pattern is exactly divisible by G, where G is called "Generator" with (r+1) bits and have known by both sender and receiver.
  - Receiver: It divides the received (d+r) bits by G. If the remainder is zero, the receiver knows the data is correct; otherwise, it is corrupted during the transmission.



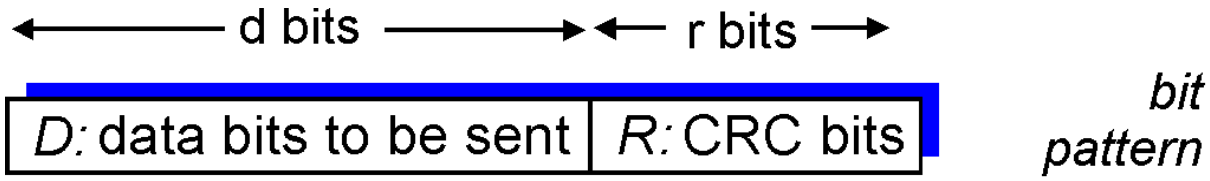
The calculation of CRC Code R

- CRC "Generator" G has r+1 bits of length. The left most bit is 1.
- G is known by both transmitter and receiver. It is a basis of CRC calculation.
- CRC code calculation is based on data D with d-bit and generator G with (r+1)-bit. The CRC code D is calculated based on the formula:

R=remainder of $D*2^r/G$

- $D*2^r$ is the appending of r bits 0 to data bits. For example

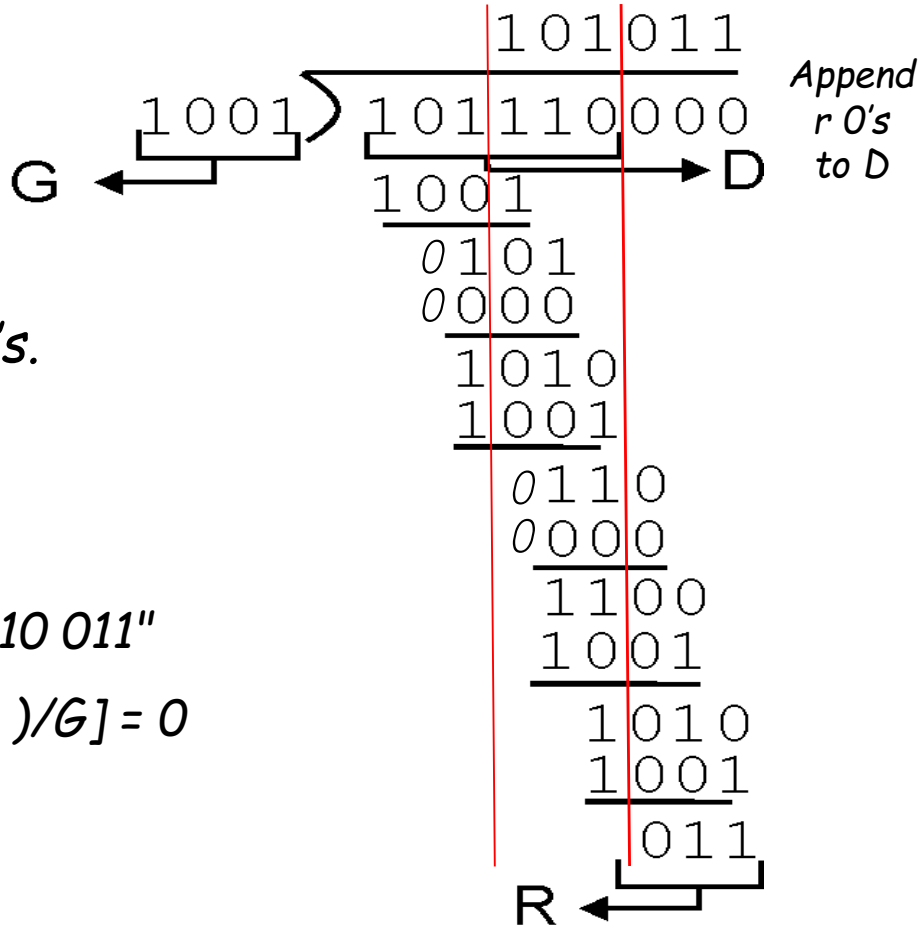
D=$(100101)_2$, and r=3, therefore, $D*2^r=(100101000)_2$



Example

- Transmitter wants to send data bits D=101110
- Generator is the (r+1)-bit G=1001.
- Calculate CRC code R (r-bits) using the formula

  R=the remainder of $D*2^r/G$
- Send data bits with CRC code appended
- Receiver side:
  - Verify the correct transmission by calculating a division for the total received data (including CRC code) by the generator.
    - Zero remainder: No error!
    - non-zero remainder: error detected!



The "CRC" = R, may have leading 0's.
Keep them.

*Transmit: D*2^r XOR R e.g., "101110 011"*

*At receiver: remainder[(D*2^r XOR R)/G] = 0*

Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

5.3 multiple access protocols

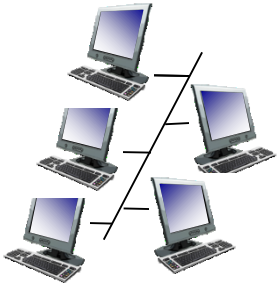
5.4 LANs

- addressing, ARP
- Ethernet
- switches

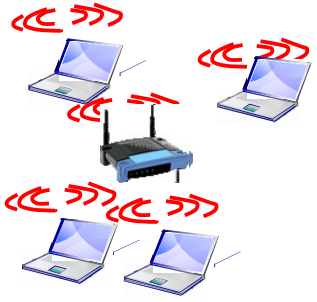
Multiple access links, protocols

two types of "links":

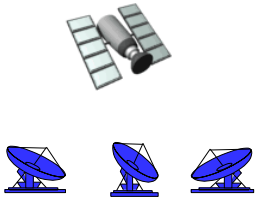
- point-to-point
  - A single node at one end of the link and a single node at other end of the link.
- *broadcast (shared wire or medium)*
  - Multiple nodes share a same, single link（collision）
  - old-fashioned Ethernet, 802.11 wireless LAN
- Multiple access problem: how to coordinate the access of multiple nodes to a broadcast link.



shared wire (e.g., cabled Ethernet)



shared RF (e.g., 802.11 WiFi)



shared RF (satellite)

humans at a cocktail party (shared air, acoustical)

Multiple access protocols (MAC)

- A human analogy: A classroom
  - Teacher(s) and student(s) share the same, single, broadcast channel
- The problem: who gets to talk? i.e., how to arrange the usage of the broadcast channel.
  - Collision occurs if multiple persons talk at the same time. Nobody can hear clearly.
  - collision if node receives two or more signals at the same time
- A mechanism or protocol is needed to control(or to organize or to schedule) the usage of broadcast channel among multiple persons------ Multiple Access Control (MAC) protocol
- Basic principles for sharing the broadcast channel
  - Raise your hand if you have a question
  - Don't interrupt when someone is speaking
  - Pay attention when someone else is speaking
  - Given everyone a chance to speak

Multiple access protocols (MAC)

- MAC protocol: it is designed to *coordinate* the transmission of different nodes in order to minimize/avoid *collision*.
- MAC protocol is to solve
  - **WHO** is going to use the link?
  - **WHEN** is the link going to be used?
  - **HOW** long is the channel used by a node?
- Goal: efficient, fair, simple, decentralized
  - Efficient and fair
    - When one node wants to transmit, it can send at rate R, where R is the rate of broadcast link.
    - When N nodes want to transmit, each can send at average rate R/N
  - Simple
    - Simple and easy to implement
  - decentralized
    - no special node is needed to coordinate transmissions
    - no synchronization among all nodes (in terms of clocks, slots)

MAC protocols: taxonomy

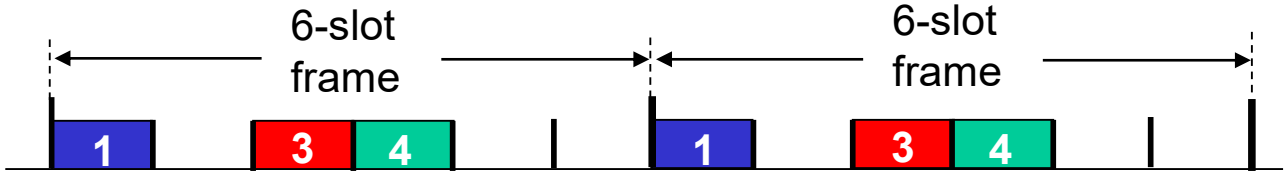
three broad classes:

- *channel partitioning protocols*
  - divide channel into smaller "pieces" (time slots, frequency, code)
  - allocate piece to node for exclusive use (TDMA, FDMA, CDMA)
- *random access protocols*
  - channel not divided, allow collisions
  - A transmitting node always transmits at the full rate of the links.
  - "recover" from collisions
  - E.g. slotted aloha
- *"taking turns" protocols*
  - nodes take turns to access the link, but nodes with more to send can take longer turns
  - No collision, but each node may take a long time to wait for its turn when there are many nodes.

Channel partitioning MAC protocols: TDMA

TDMA: time division multiple access

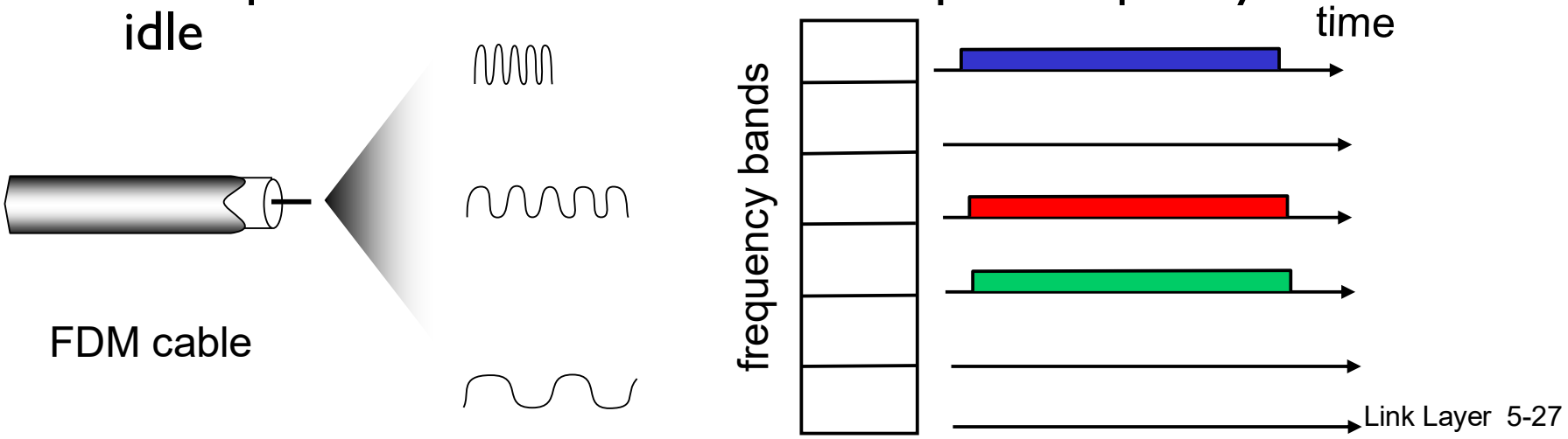
- Time is divided into frames and each frame is further divided into N time slots.
- Each time slot is assigned to one of the N nodes.
- Whenever a node has a packet to send, it transmits the packets during its assigned time slots at the rate of this link R. unused slots go idle
- Different nodes are separated in the time domain.
- example: 6-station LAN, 1,3,4 have pkt, slots 2,5,6 idle



Channel partitioning MAC protocols: FDMA

FDMA: frequency division multiple access

- Frequency band is divided into N different parts.
- Each frequency part is assigned to one of the N nodes.
- Each user transmits with no limitation in time, but use only the frequency assigned to each user.
- Different users are separated in the frequency domain.
- unused transmission time in frequency bands go idle
- example: 6-station LAN, 1,3,4 have pkt, frequency bands 2,5,6 idle



TDMA and FDMA

Advantages

- Simple
- Eliminate collision
- Provide fair allocation

Disadvantages

- Limit the average rate to R/N
- Resource waste due to the inactive nodes.

Random access protocols

- when node has packet to send
  - transmit at full channel data rate R at a random time
  - no *a priori* coordination among nodes
- two or more transmitting nodes ➜ "collision",
- random access MAC protocol specifies:
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed retransmissions)
- examples of random access MAC protocols:
  - slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA

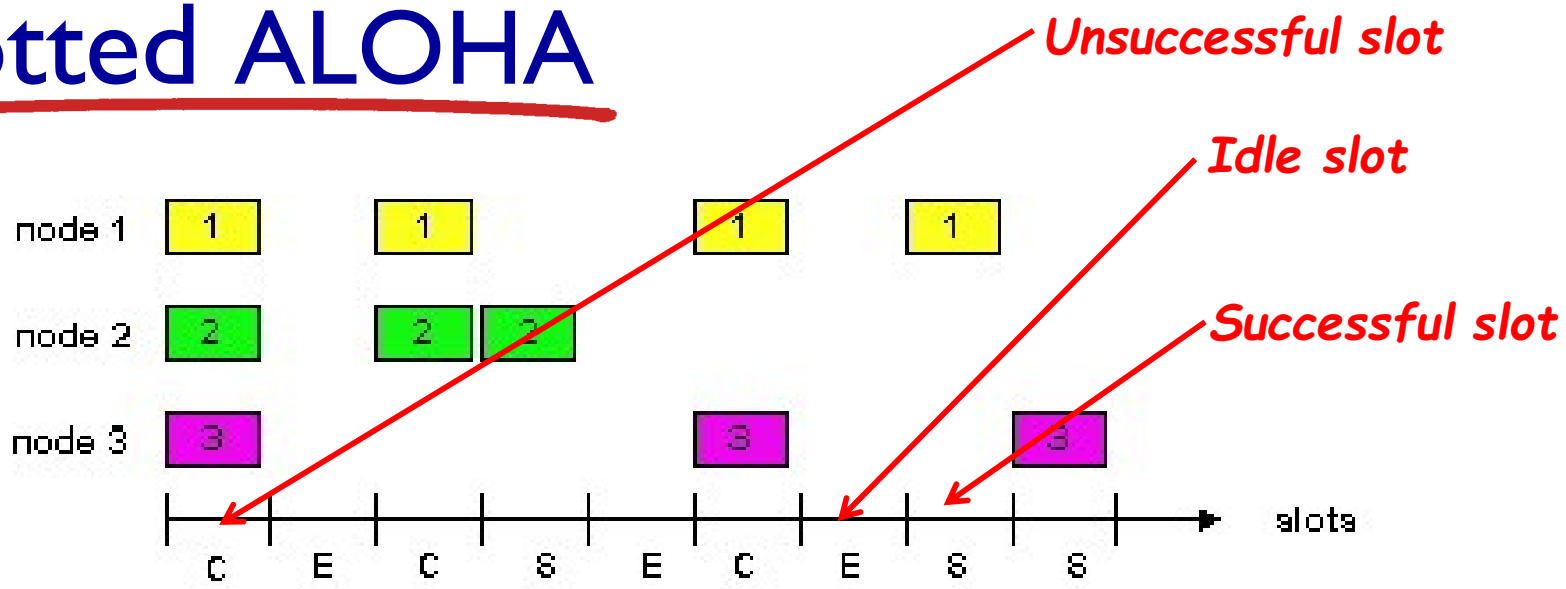
assumptions:

- all frames consist of exactly fixed-size bits
- 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 prob. p until success

Slotted ALOHA



Pros:

- single active node can continuously transmit at full rate of channel
- highly decentralized:
  - nodes detect collision independently.
  - Node decides when to retransmit independently.
- simple

Cons:

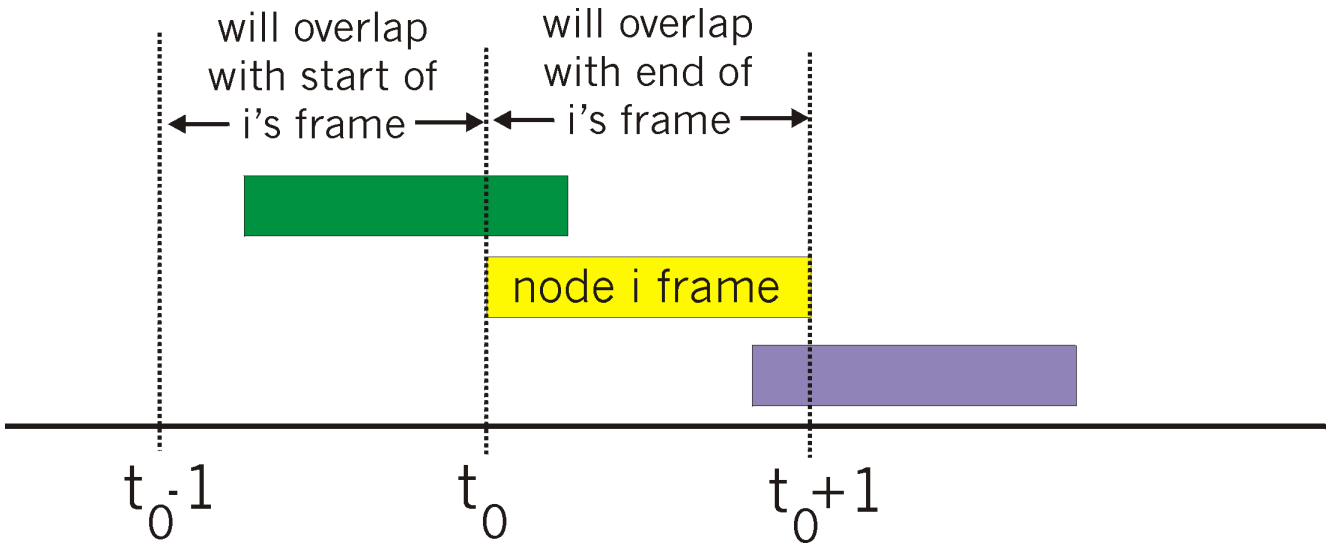
- collisions, wasting slots
- idle slots
- clock synchronization

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

Pure (unslotted) ALOHA

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

even *worse* than slotted Aloha!



CSMA (carrier sensing multiple access)

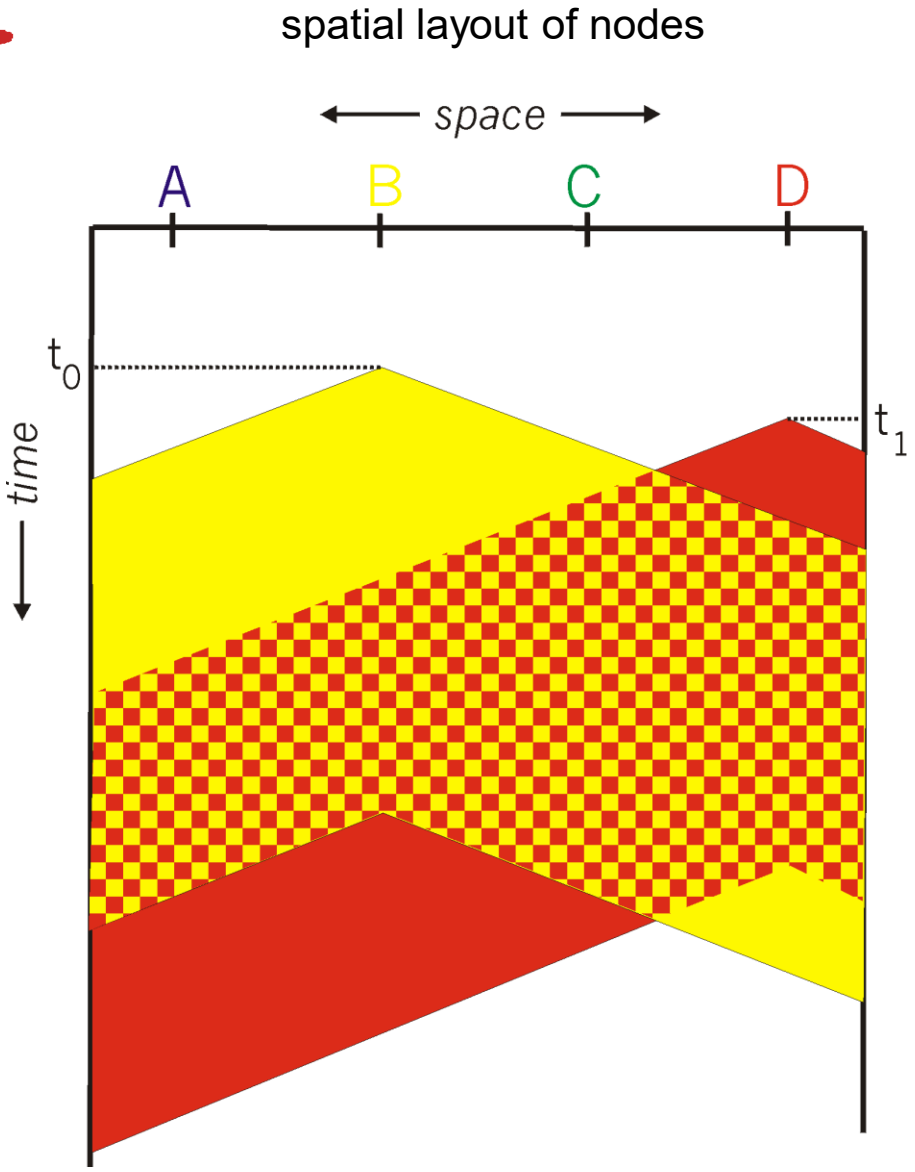
In order to reduce the amount of collision, we introduce carrier sensing multiple access with collision detection (CSMA/CD).

CSMA:

- CSMA: listen before transmit. If channel is sensed busy, defer transmission; do transmission only if channel is sensed to be idle.
- Note that collisions may still exist since two nodes may sense the channel idle at the same time
- In case of collision, the entire frame transmission time is wasted.

CSMA collisions

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



CSMA/CD (collision detection)

Collision Detection(CD): A transmitting node listens to the channel while it is transmitting. If it detects that another node is transmitting at the same time, it stops transmitting and uses some protocol to determine when it should do the next attempt to transmit.

CSMA/CD: carrier sensing, deferral as in CSMA

- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage

- collision detection:
  - easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - difficult in wireless LANs: received signal strength overwhelmed by local transmission strength
- human analogy: the polite conversationalist

"Taking turns" MAC protocols

- **Full rate**: when only one node is active, it can have the full rate of the link.
- **Fairness**: when N nodes are active, each active node has a throughput of nearly R/N.

channel partitioning MAC protocols----have the second property

- 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!

Random access MAC protocols—have the first property

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

"Taking turns" protocols---have both properties

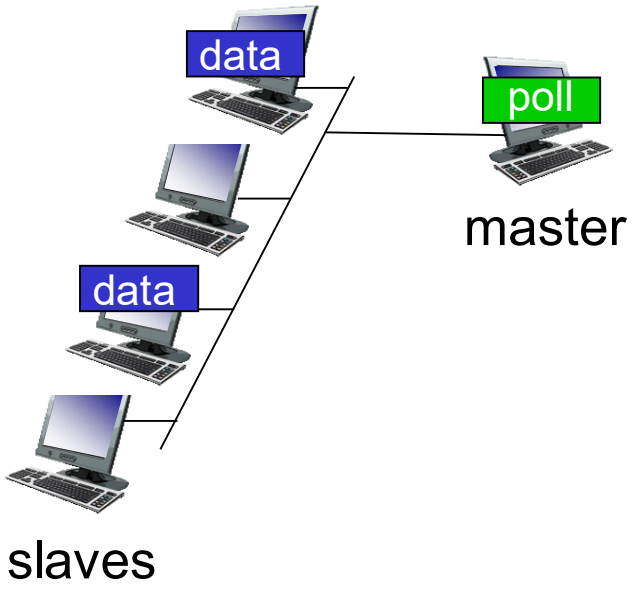
- It can achieve both fairness and full rate, at the expense of some extra overhead

"Taking Turns" MAC Protocol

- *Two kinds of "taking turns" protocols*
  - *Polling protocols*
  - *Token passing protocol*

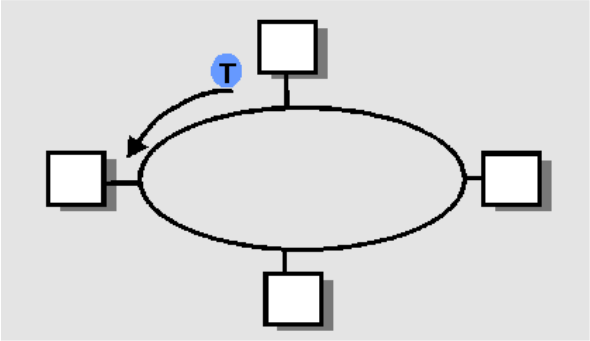
"Taking turns" MAC protocols

- Polling protocol
  - A node is designated as a master node.
  - Master node polls each of nodes in a round-robin fashion and to "invite" them to transmit in turn.
- Advantages
  - Eliminate the collision
- Disadvantages:
  - Single point of failure (master node)
  - Polling overhead
  - Latency



"Taking turns" MAC protocols

- *Token passing protocol*
  - *Token: it is a small, special-purpose frame.*
  - *Token is passed from one node to another node in a fixed order.*
  - *When a node receives the token, the node immediately forwards the token to the next node if it has no frames to send; otherwise, it holds the token and send frames at the full rate of link for a period of time, then passes the token to the next node.*
- *Advantages*
  - *Decentralized and efficient.*
- *Disadvantages:*
  - *Single point of failure (token)*
  - *Token overhead*
  - *Latency*



Summary of MAC protocols

- *channel partitioning,* by time, frequency or code
  - TDMA, FDMA, CDMA
- *random access* (dynamic),
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - carrier sensing: easy in some technologies (wire), hard in others (wireless)
  - CSMA/CD used in Ethernet
  - CSMA/CA used in 802.11
- *taking turns*
  - polling from central site, token passing
  - bluetooth, FDDI, token ring

Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

5.3 multiple access protocols

5.4 LANs

- addressing, ARP
- Ethernet
- switches

Local Area Networks (LAN)

- LAN is a computer network concentrated in a geographical area at the scale of from 10m to 1km, such as in a building or on a university campus
- MAC protocols are used in LANs to control access to the shared channel.

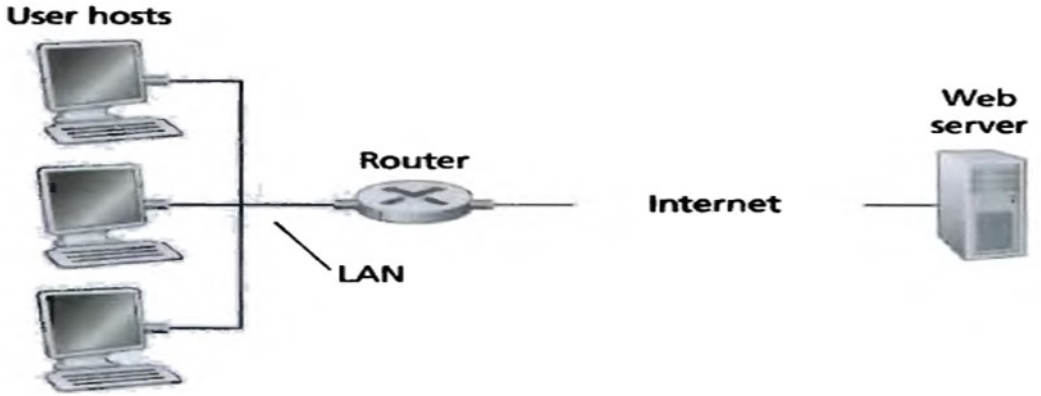


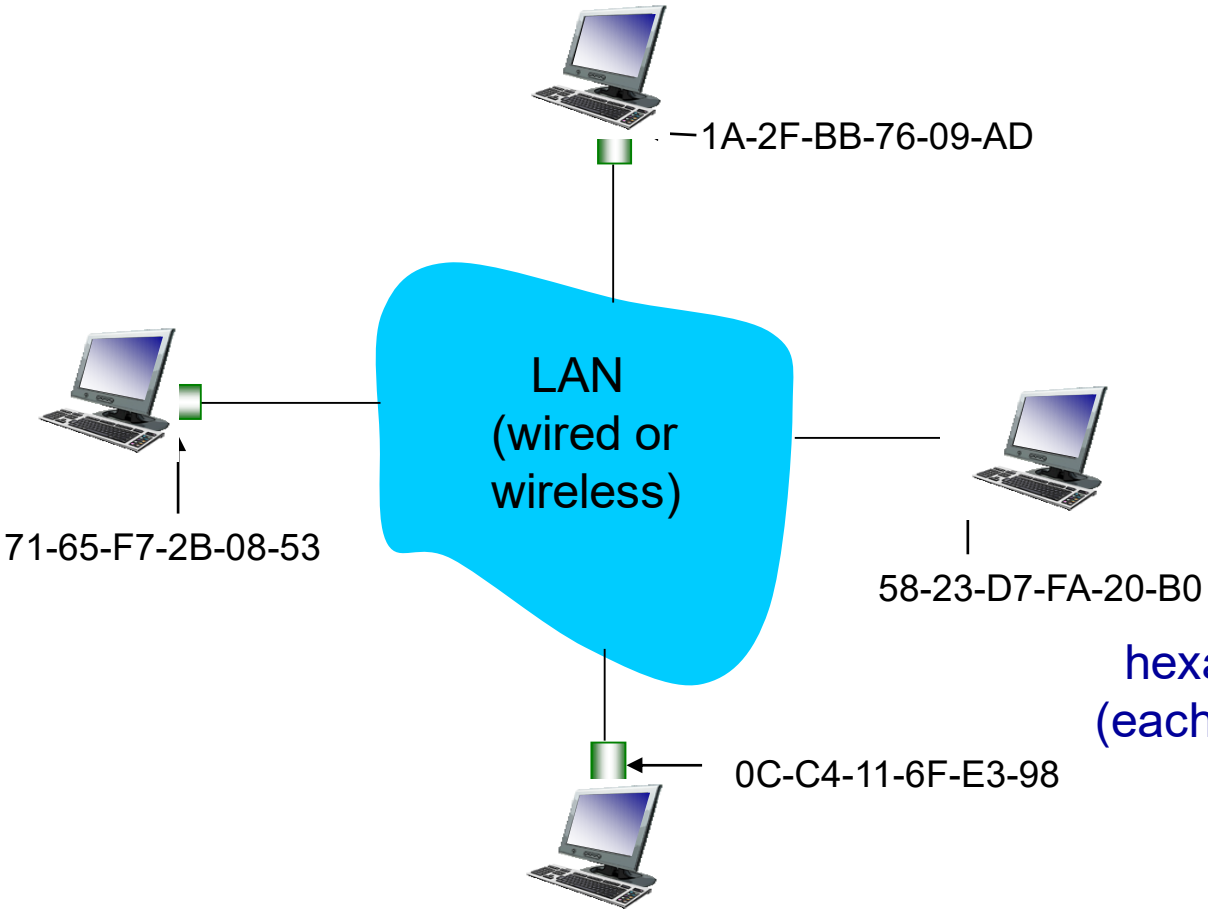
Figure 5.15 ◆ User hosts access an Internet Web server through a LAN, where the broadcast channel between a user host and the router consists of one link.

MAC addresses and ARP

- 32-bit IP address:
  - *network-layer* address for interface: : a.b.c.d/x
  - used for layer 3 (network layer) forwarding
  - DNS provides the transformation from hostname to IP address
- Each adapter on a host has a unique MAC (or LAN or physical or Ethernet) address:
  - used to direct frame from one node's LAN interface to the another node's LAN interface card (adapter card) on the local LAN (i.e., both interfaces are physically-connected)
  - 48 bit MAC address (for most LANs) burned in adapter ROM, also sometimes software settable
  - ARP(address resolution protocol) provides the service of getting the MAC address of a specified IP address.

LAN addresses and ARP

each adapter on LAN has unique *LAN* address



Broadcast address =
FF-FF-FF-FF-FF-FF

adapter

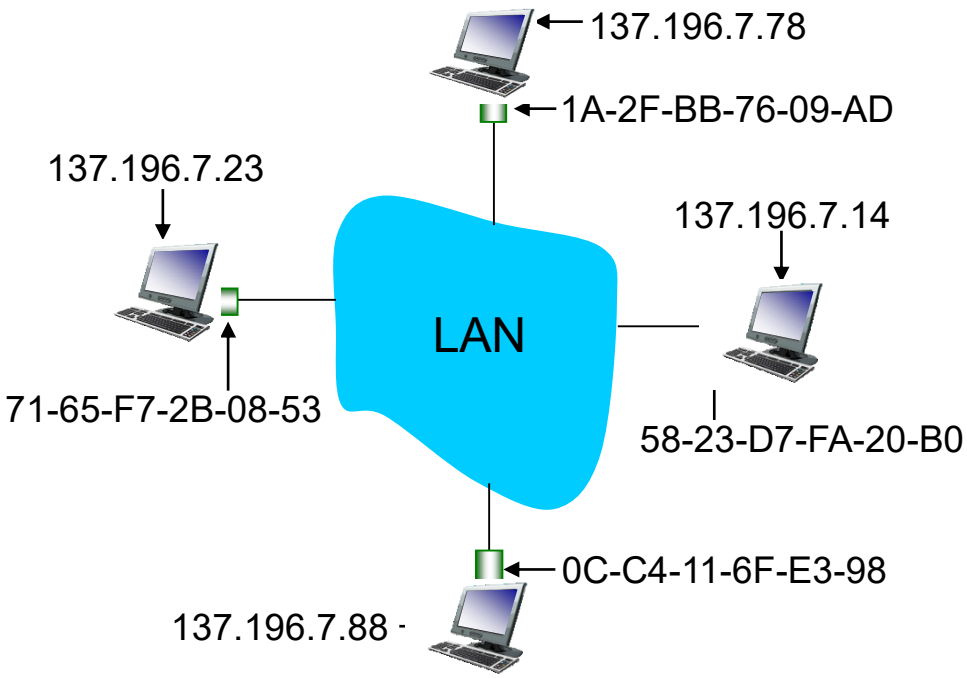
hexadecimal (base 16) notation
(each "number" represents 4 bits)

LAN addresses (more)

- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space (to assure uniqueness)
- Each adapter (i.e. LAN interface) connected to a LAN has a unique MAC address.
- analogy:
  - MAC address: like Social Security Number, is fixed
  - 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 ARP table

- ARP table: IP/MAC address mappings for some LAN nodes:

  < IP address; MAC address; TTL>

- TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

IP Address	MAC Address	TTL
222.222.222.221	88-B2-2F-54-1A-0F	13:45:00
222.222.222.223	5C-66-AB-90-75-B1	13:52:00

ARP protocol: same LAN

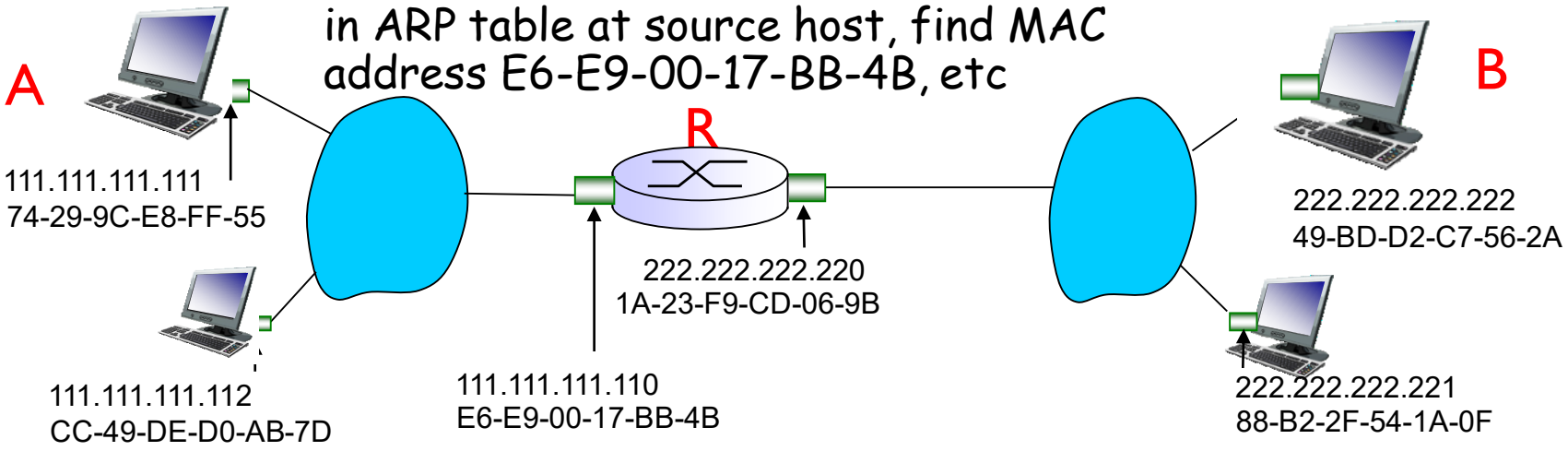
- A wants to send datagram to B
  - B's MAC address not in A's ARP table.
- A broadcasts ARP query packet, containing B's IP address
  - broadcast MAC address = FF-FF-FF-FF-FF-FF
  - all nodes on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
  - frame sent to 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*

Addressing: routing to another LAN

walkthrough: send datagram from A to B via R

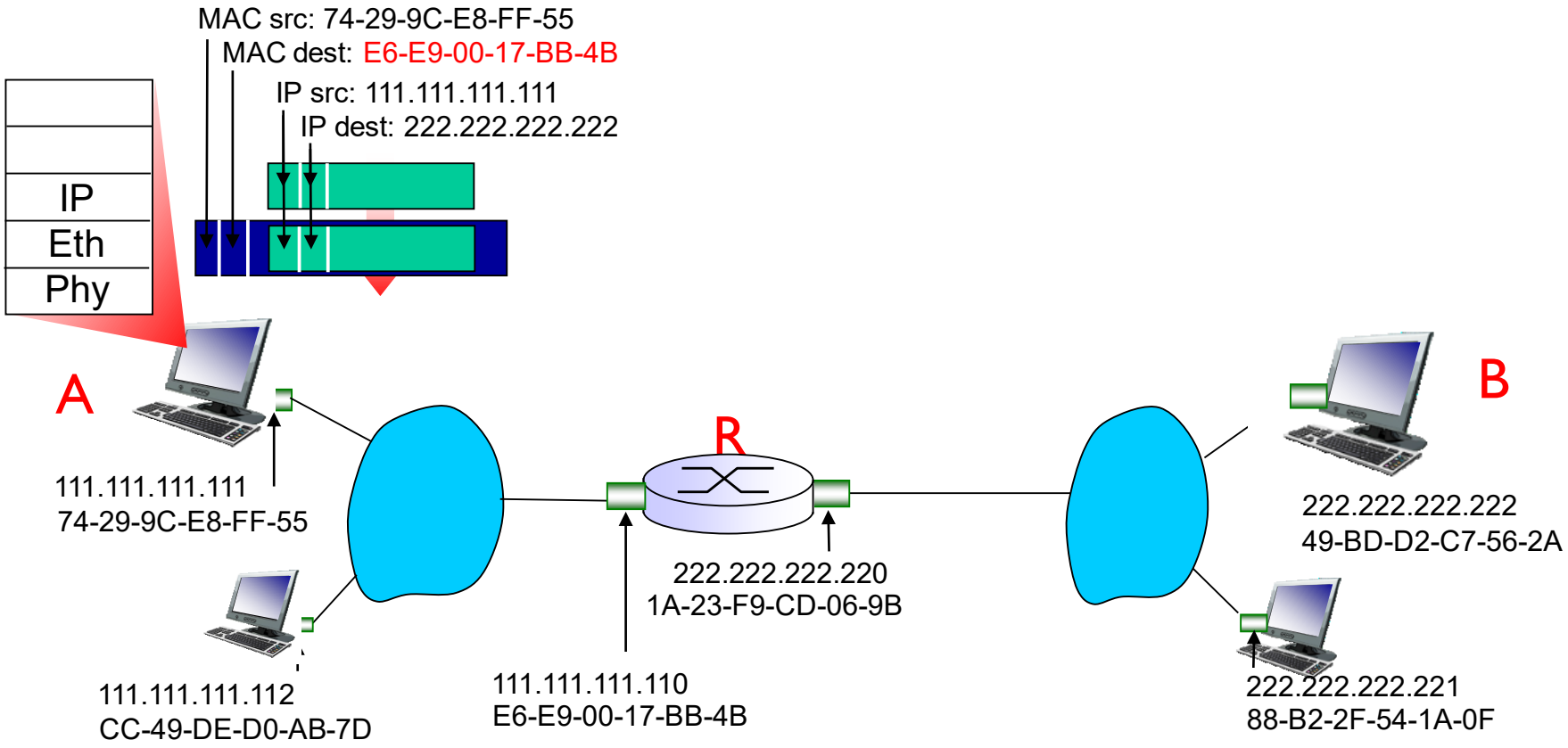
- focus on addressing – at IP (datagram) and MAC layer (frame)
- assume A knows B's IP address
- assume A knows IP address of first hop router, R (how?)
- assume A knows R's MAC address (how?)

two ARP tables in router R respectively
for two subnets (i.e., LAN)
in routing table at source host, find
router 111.111.111.110
in ARP table at source host, find MAC
address E6-E9-00-17-BB-4B, etc



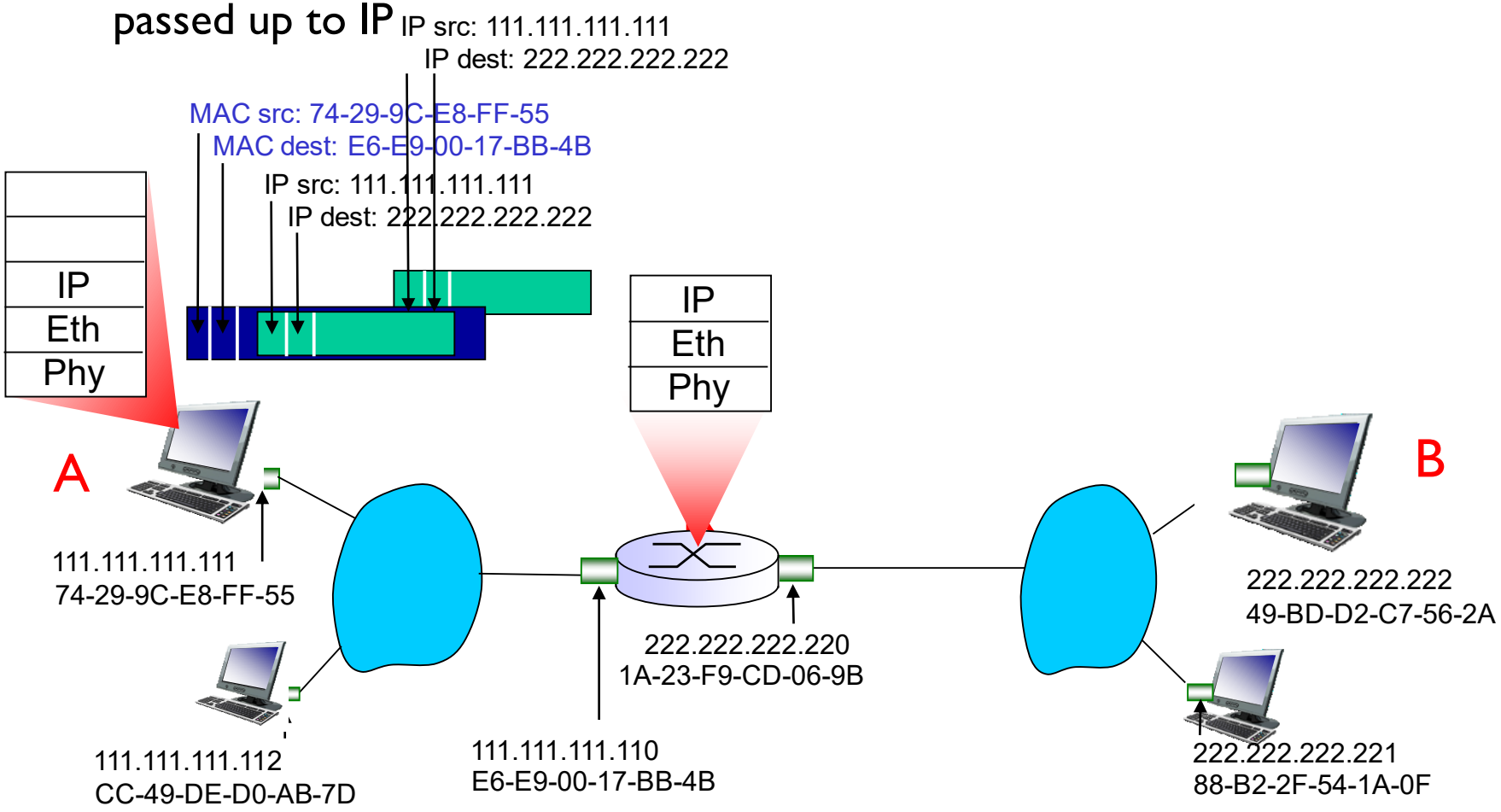
Addressing: routing to another LAN

- A creates IP datagram with IP source A, destination B
- A uses ARP to get R's MAC address for 111.111.111.110
- A creates link-layer frame with R's MAC address as dest, frame contains A-to-B IP datagram



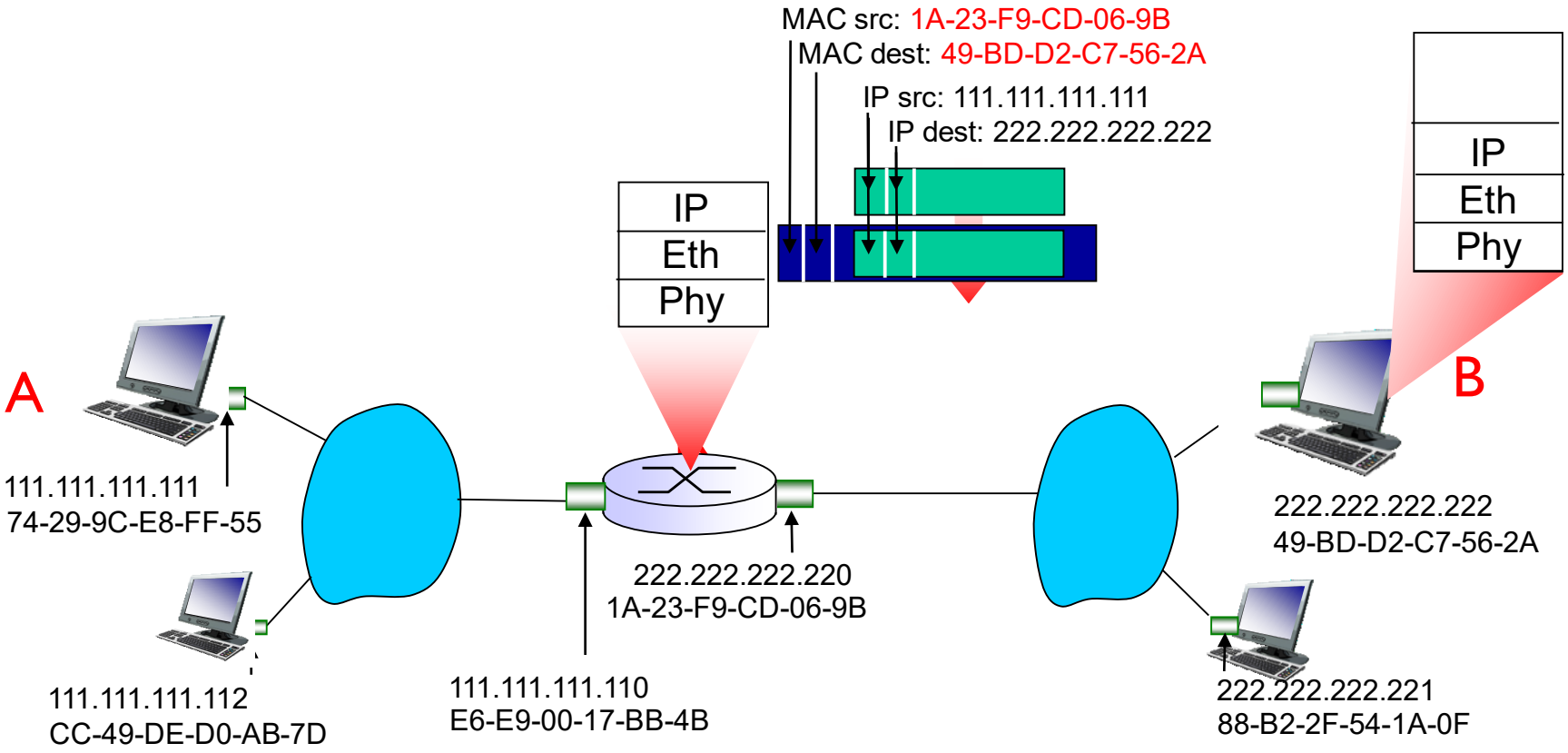
Addressing: routing to another LAN

- A's Adapter sends frame from A to R
- frame received at R, datagram removed, see its destination to B, passed up to IP



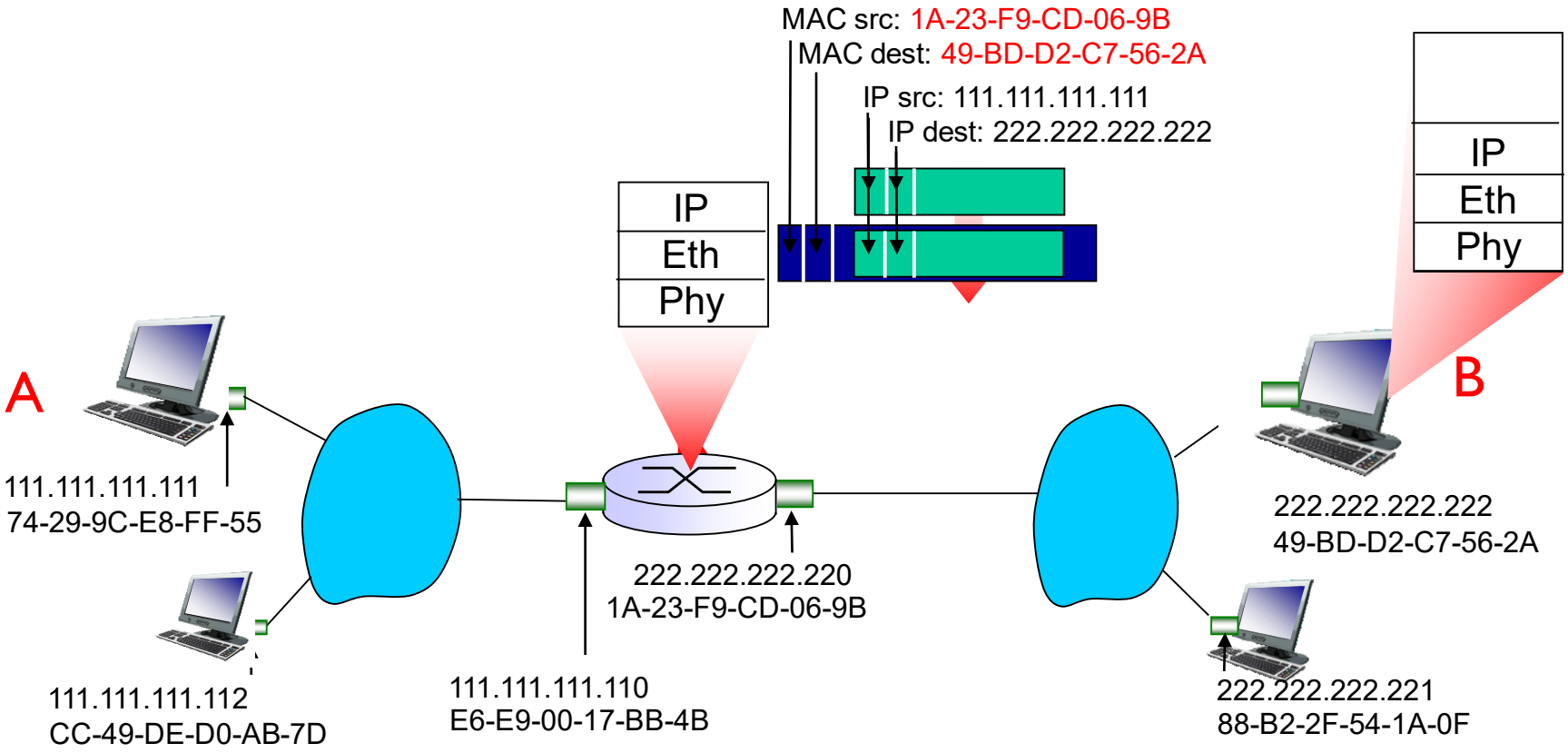
Addressing: routing to another LAN

- 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



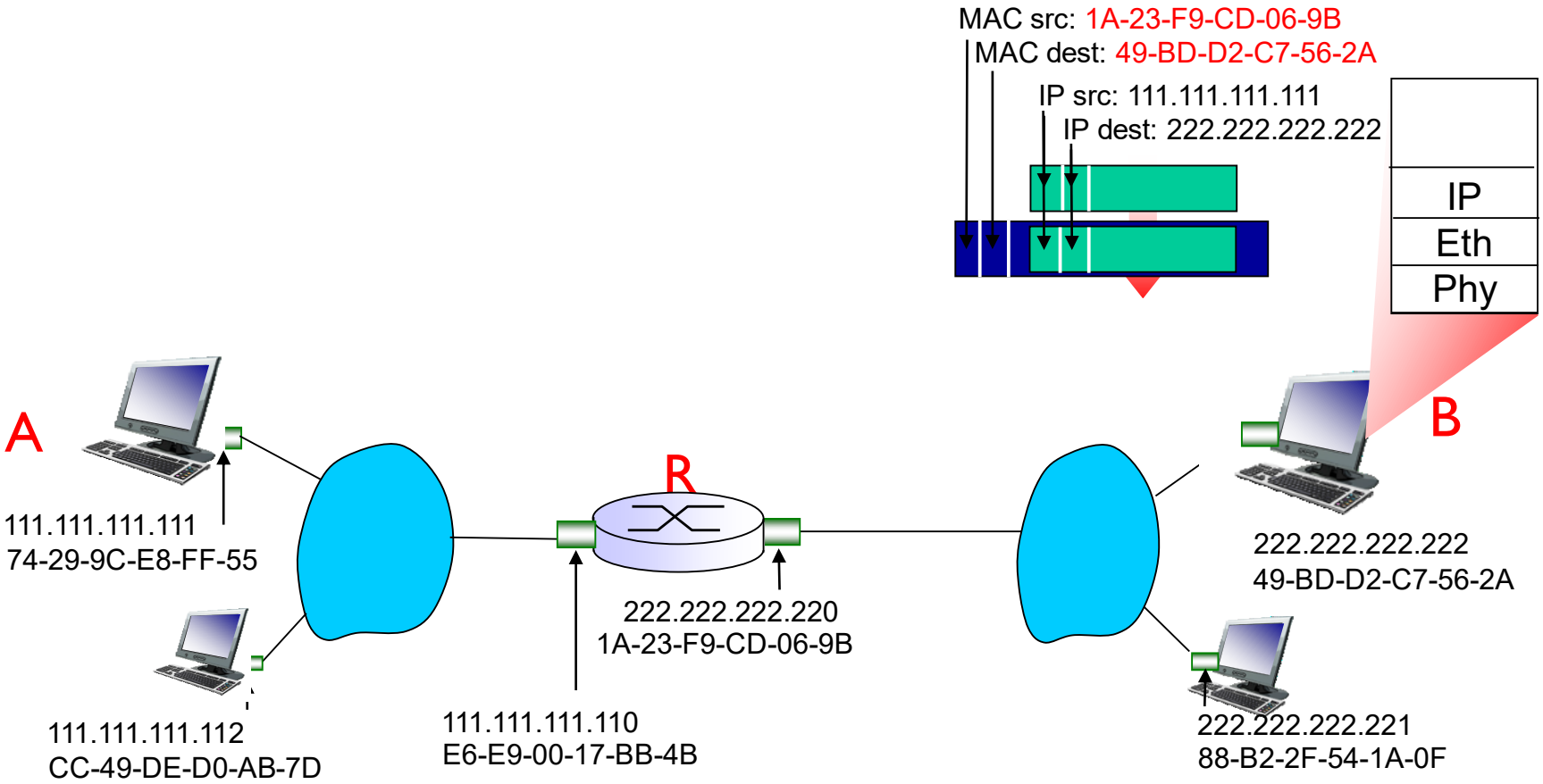
Addressing: routing to another LAN

- 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



Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

5.3 multiple access protocols

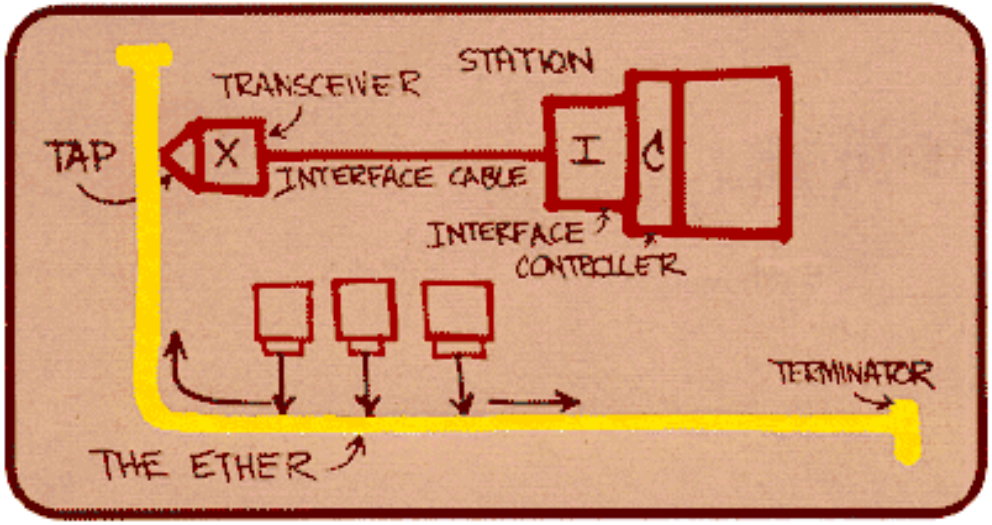
5.4 LANs

- addressing, ARP
- Ethernet
- switches

Ethernet

"dominant" wired LAN technology:

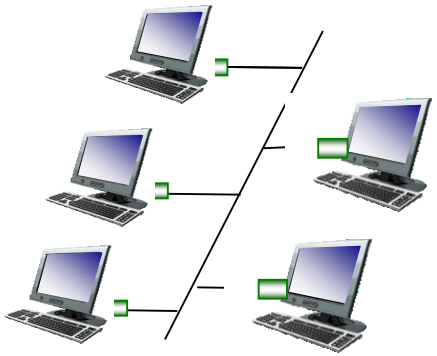
- cheap $20 for NIC
- first widely used LAN technology
- simpler, cheaper than token LANs and ATM
- kept up with speed race: 10 Mbps – 10 Gbps



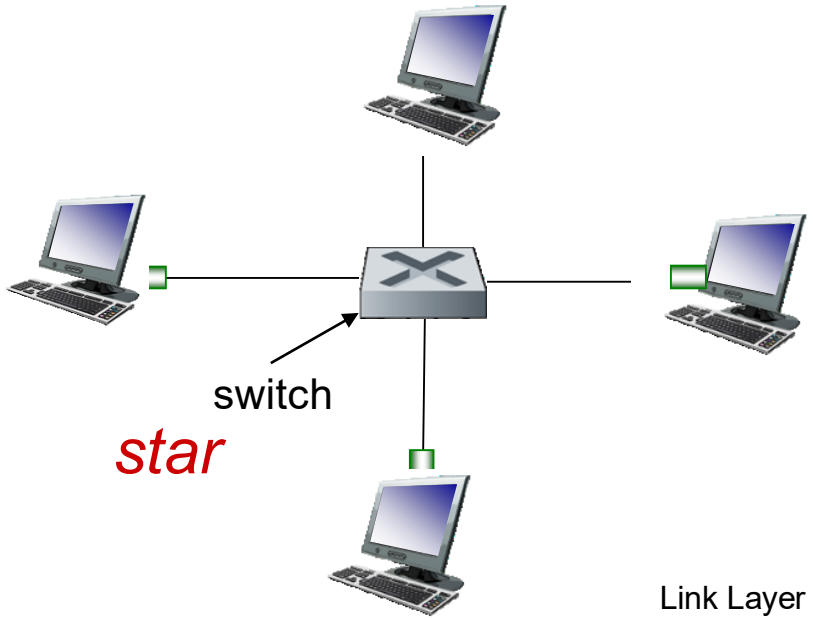
Metcalfe's Ethernet sketch

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



star

Ethernet frame structure

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

preamble:

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

Ethernet frame structure (more)

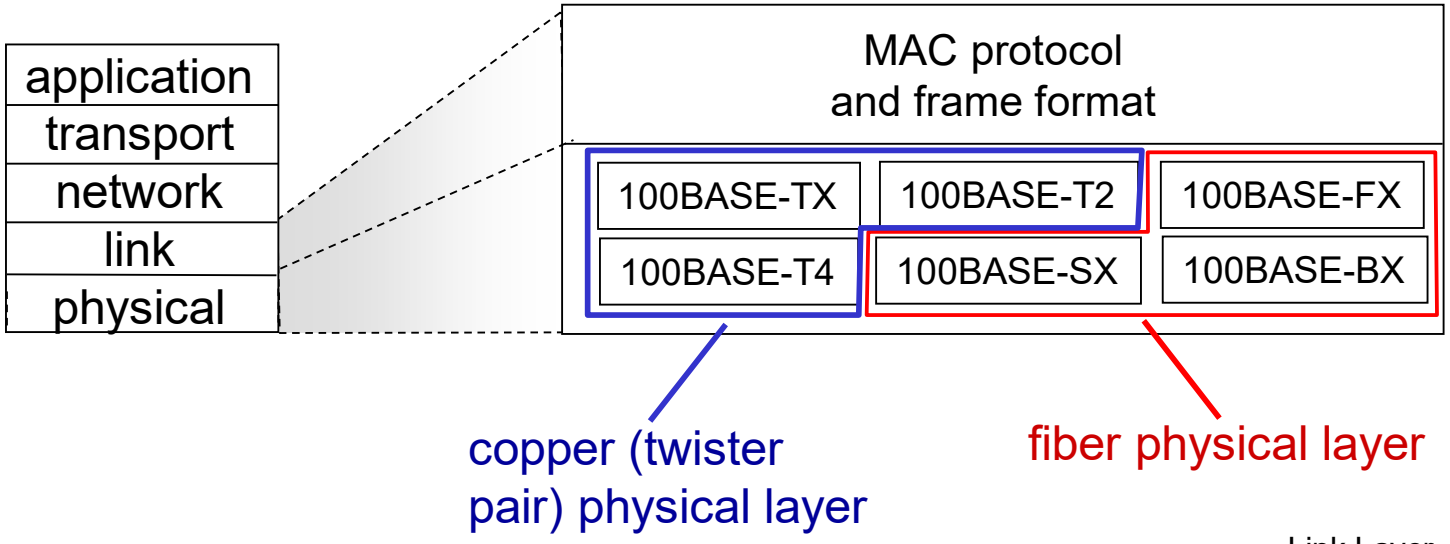
- *addresses:* 6 byte source, destination MAC addresses
  - if adapter receives frame with matching destination address, or with broadcast address (e.g. ARP packet), it passes data in frame to network layer protocol
  - otherwise, adapter discards frame
- *type:* indicates higher layer protocol (mostly 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 doesnt 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 wth 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, 1Gbps, 10G bps
  - different physical layer media: fiber, cable



Link layer, LANs: outline

5.1 introduction, services

5.2 error detection, correction

5.3 multiple access protocols

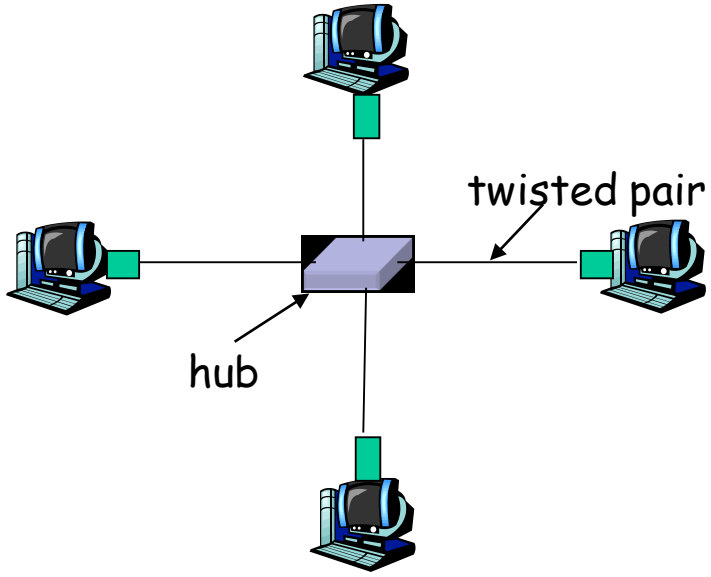
5.4 LANs

- addressing, ARP
- Ethernet
- switches

Hubs

… physical-layer ("dumb") repeaters:

- bits coming in one link go out *all* other links at same rate
- all nodes connected to hub can collide with one another
- no frame buffering
- no CSMA/CD at hub: host NICs detect collisions

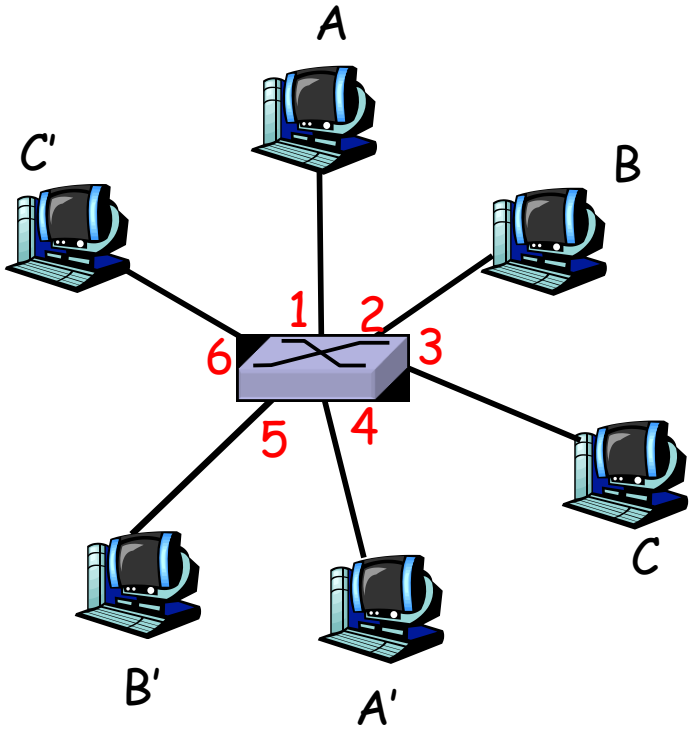


Ethernet switch

- 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 are unaware of presence of switches
- *plug-and-play, self-learning*
  - switches do not need to be configured

Switch: allows *multiple* simultaneous transmissions

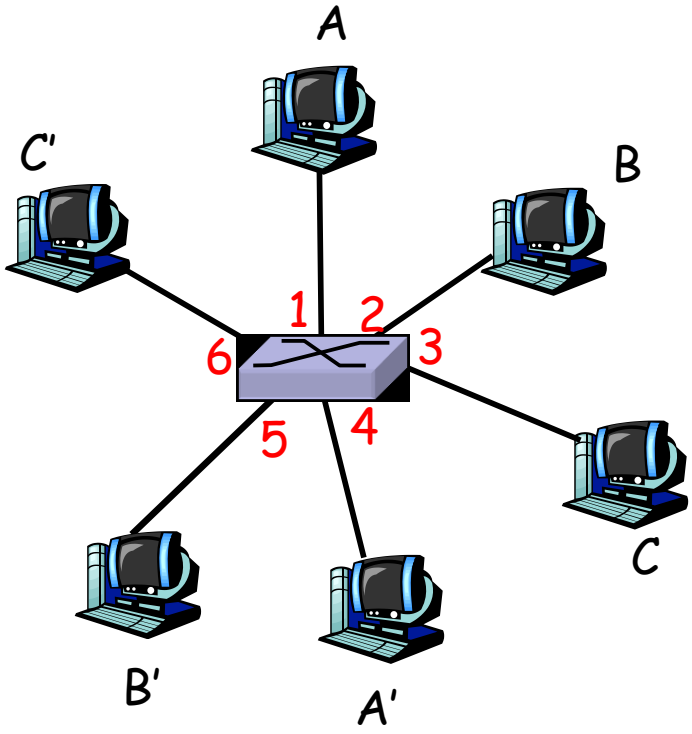
- hosts have dedicated, direct connection to switch
- switches buffer packets
- Ethernet protocol used on *each* incoming link, but no collisions; full duplex
  - each link is its own collision domain
- *switching:* A-to-A' and B-to-B' simultaneously, without collisions
  - not possible with dumb hub



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

Switch Table

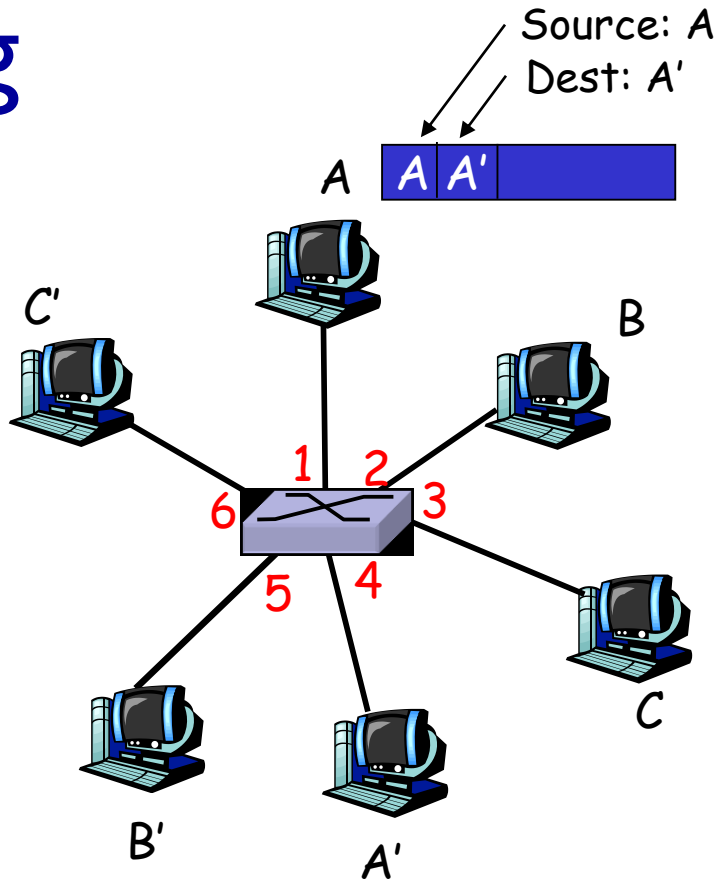
- Q: how does switch know that A' reachable via interface 4, B' reachable via interface 5?
- A: each switch has a switch table, each entry:
  - (MAC address of host, interface to reach host, time stamp)
- looks like a routing table!
- Q: how are entries created, maintained in switch table?
  - something like a routing protocol?



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

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: frame filtering/forwarding

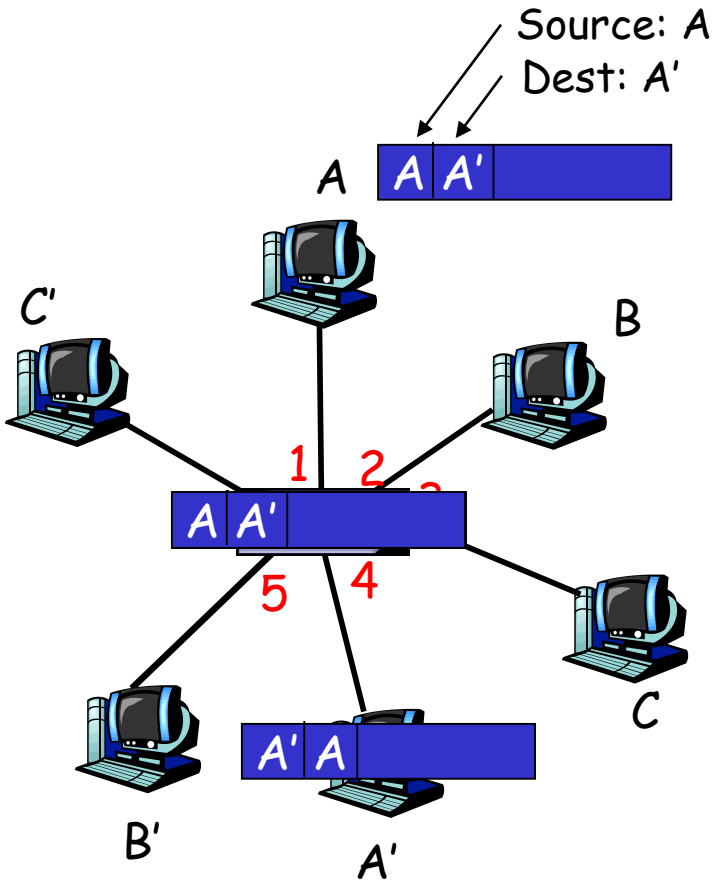
When frame received:

1. record link associated with sending host
2. index switch table using MAC dest address
3. **if** entry found for destination
   then {
   if dest on segment from which frame arrived
   then drop the frame
   else forward the frame on interface indicated
   }
   else flood

forward on all but the interface on which the frame arrived

Self-learning, forwarding: example

- frame destination unknown: *flood*
- destination A location known: *selective send*

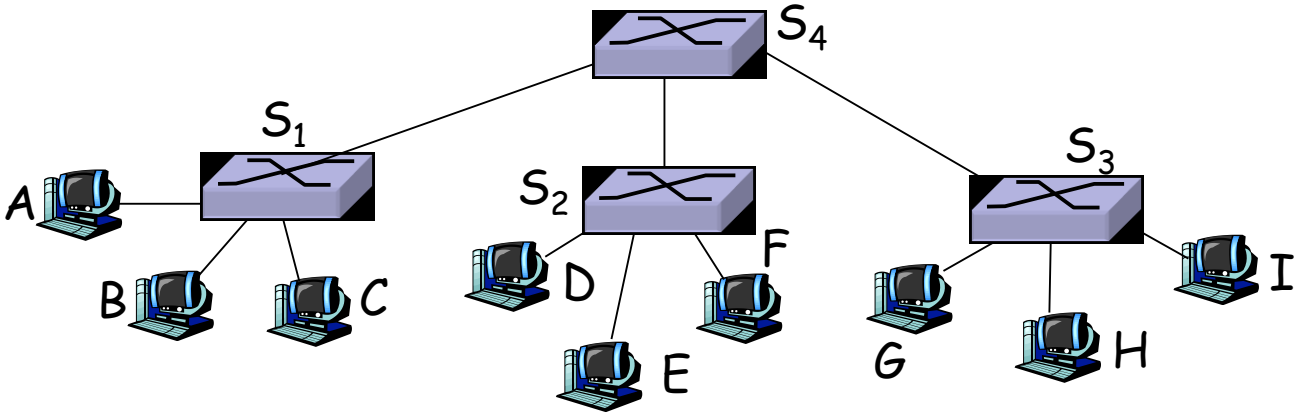


MAC addr	interface	TTL
A	1	60
A'	4	60

Switch table (initially empty)

Interconnecting switches

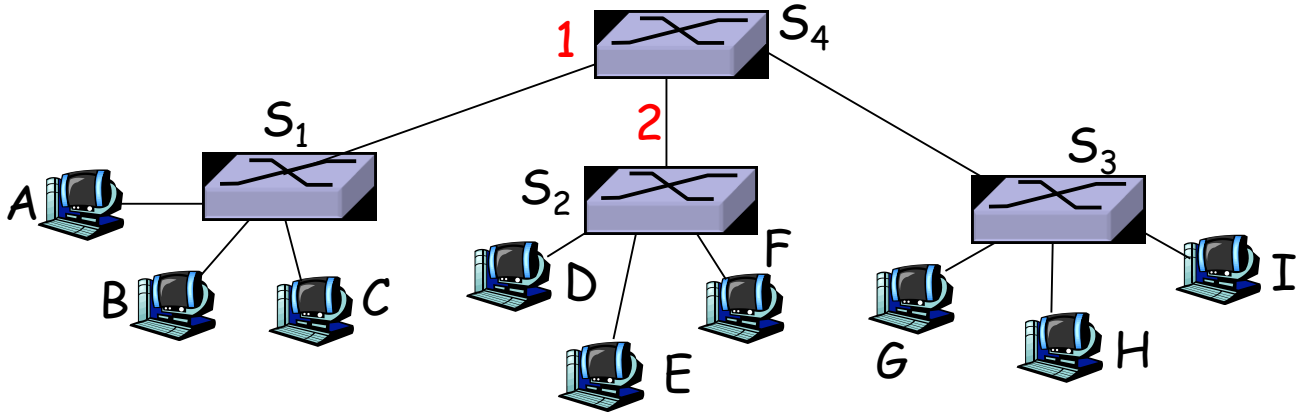
- switches can be connected together



- Q: sending from A to G - how does S_1 know to forward frame destined to F via S_4 and S_3?
- A: self learning! (works exactly the same as in single-switch case!)

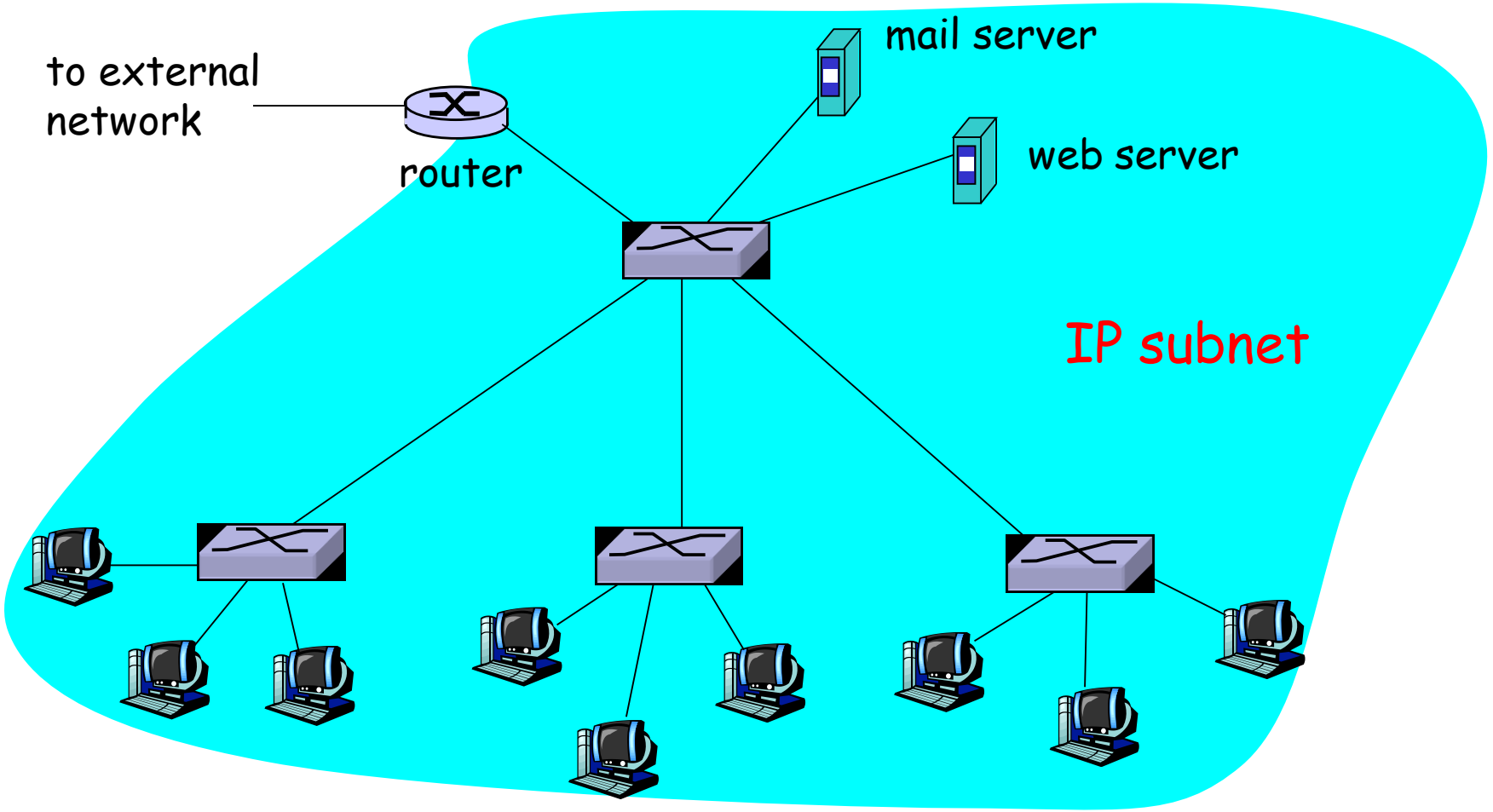
Self-learning multi-switch example

Suppose C sends frame to I, I responds to C



- Q: show switch tables and packet forwarding in S_1, S_2, S_3, S_4

Institutional network



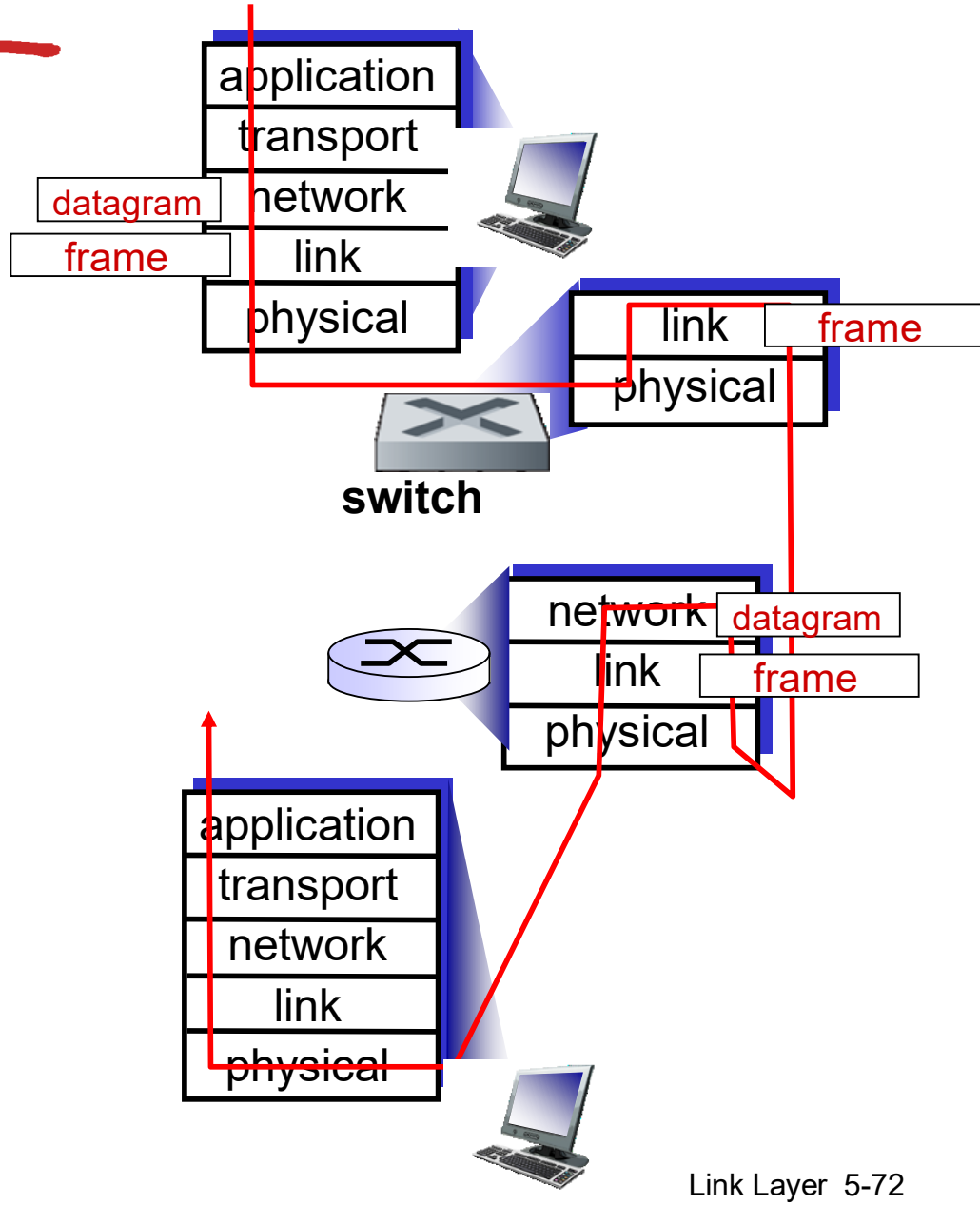
Switches vs. routers

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



Chapter 5: Summary

- principles behind data link layer services:
  - error detection, correction
  - sharing a broadcast channel: multiple access
  - link layer addressing
- instantiation and implementation of various link layer technologies
  - Ethernet
  - switched LANS