Link layer, LANs: outline

Link layer: introduction

terminology:

- hosts and routers: nodes
- communication channels that connect adjacent nodes along communication path: links
  - wired links
  - wireless links
- layer-2 packet: frame, encapsulates datagram

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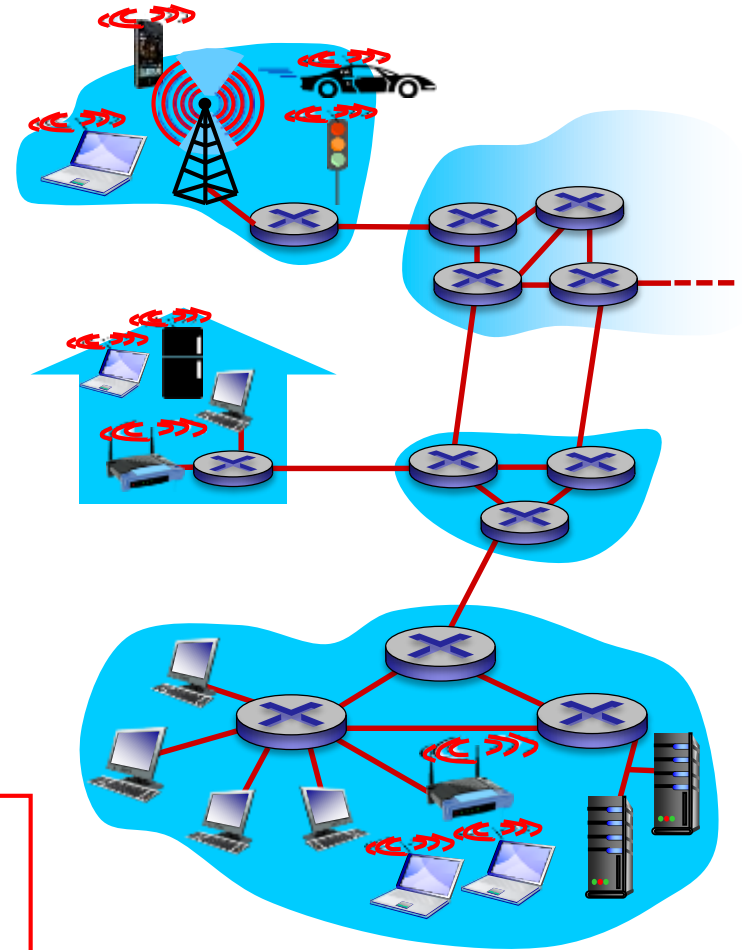
Link layer: context

- datagram transferred by different link protocols over different links:
  - e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link
- each link protocol provides different services

transportation analogy:

- trip from Victoria to Vancouver
  - limo: Victoria to Swartz Bay
  - ferry: Swartz Bay to Tsawwassen
  - texi: Tsawwassen to Vancouver
- tourist = datagram
- transport segment = communication link
- transportation mode = link layer protocol
- travel agent (or your plan) = routing algorithm

Link layer services

- *framing, link access:*
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - "MAC" addresses used in frame headers to identify source, destination
    - different from IP address!
- *reliable delivery between adjacent nodes*
  - retransmissions
  - seldom used on low bit-error link (fiber, some twisted pair)
  - wireless links: high error rates
    - *Q:* why both link-level and end-end reliability?

Link layer services (more)

- *error detection:*
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
- *error correction:*
  - receiver identifies *and corrects* bit error(s) without resorting to retransmission
- *half-duplex and full-duplex*
  - with half duplex, nodes at both ends of link can transmit, but not at same time

Link layer, LANs: outline

2.1 introduction, services

2.2 error detection, correction

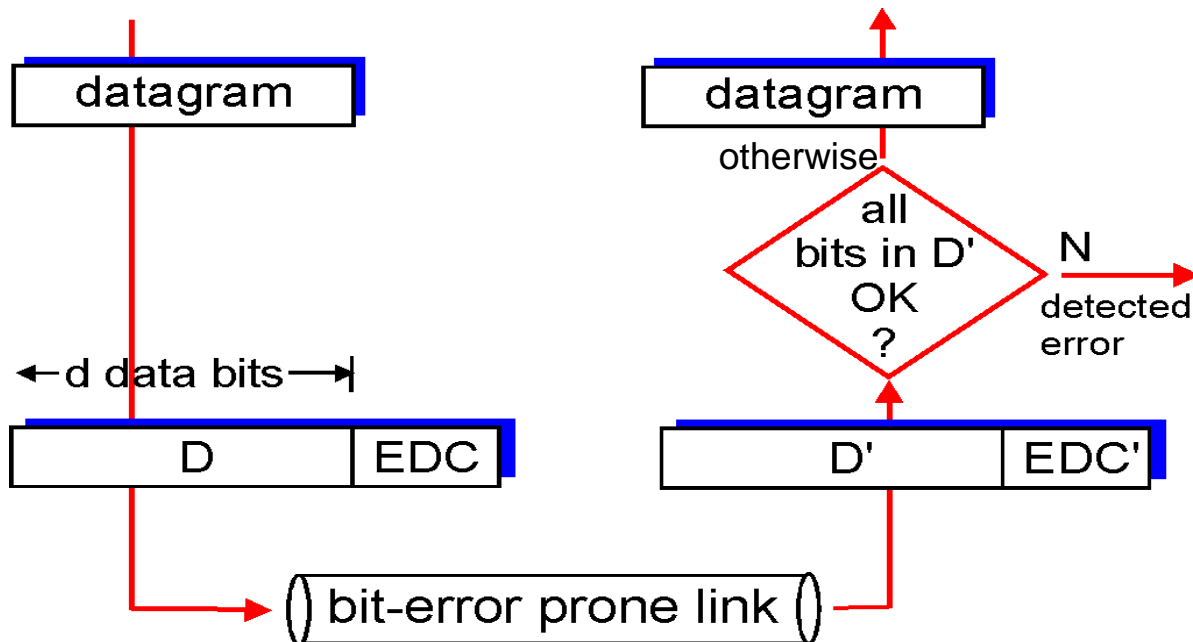2.3 multiple access protocols

2.4 LANs
- addressing, ARP
- Ethernet
- switches

Error detection

EDC= Error Detection and Correction bits (redundancy)
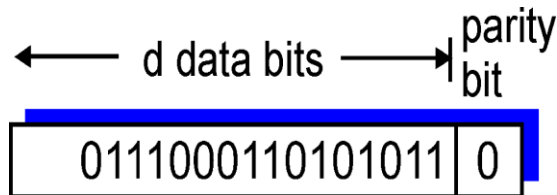D   = Data protected by error checking, may include header fields

- Error detection not 100% reliable!
    - protocol may miss some errors, but rarely
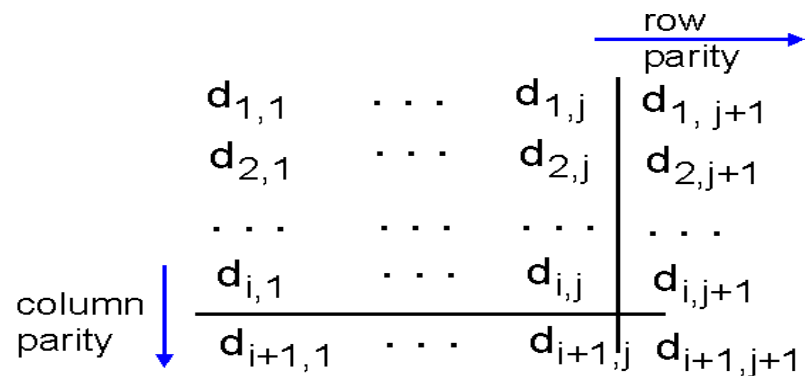   - larger EDC field yields better detection and correction

Parity checking

single bit parity:

- *d*etect single bit errors

two-dimensional bit parity:

- detect and correct single bit errors

$$
\begin{array}{cccc|c}
 & & & & \text{row parity} \\
d_{1,1} & \cdots & d_{1,j} & & d_{1,\,j+1} \\
d_{2,1} & \cdots & d_{2,j} & & d_{2,j+1} \\
\cdots & \cdots & \cdots & & \cdots \\
d_{i,1} & \cdots & d_{i,j} & & d_{i,j+1} \\
\hline
d_{i+1,1} & \cdots & d_{i+1,j} & & d_{i+1,j+1}
\end{array}
$$

column parity

```
1 0 1 0 1|1          1 0 1 0 1|1
1 1 1 1 0|0          1 0 1 1 0 0   parity error
0 1 1 1 0|1          0 1 1 1 0|1
0 0 1 0 1|0          0 0 1 0 1|0
```

no errors   parity error

correctable single bit error

8

Internet checksum

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 checksum field (e.g., TCP)

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?*

Cyclic redundancy check

- more powerful error-detection coding
- view data bits, D, as a binary number
- choose r+1 bit pattern (generator), G
- goal: choose r CRC bits, R, such that
  - <D,R> exactly divisible by G (modulo 2)
  - receiver knows G, divides <D,R> by G. If non-zero remainder: error detected!
  - can detect all burst errors less than r+1 bits
- widely used in practice (Ethernet, 802.11 WiFi, ATM)

$$D * 2^r \; XOR \; R$$
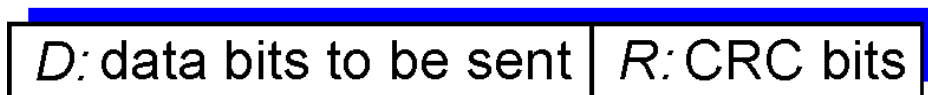
bit pattern

mathematical formula

CRC example

want:

D·2^r XOR R = nG

equivalently:

D·2^r = nG XOR R

equivalently:

if we divide D·2^r by G, want remainder R to satisfy:

$$R = remainder[\frac{D \cdot 2^r}{G}]$$
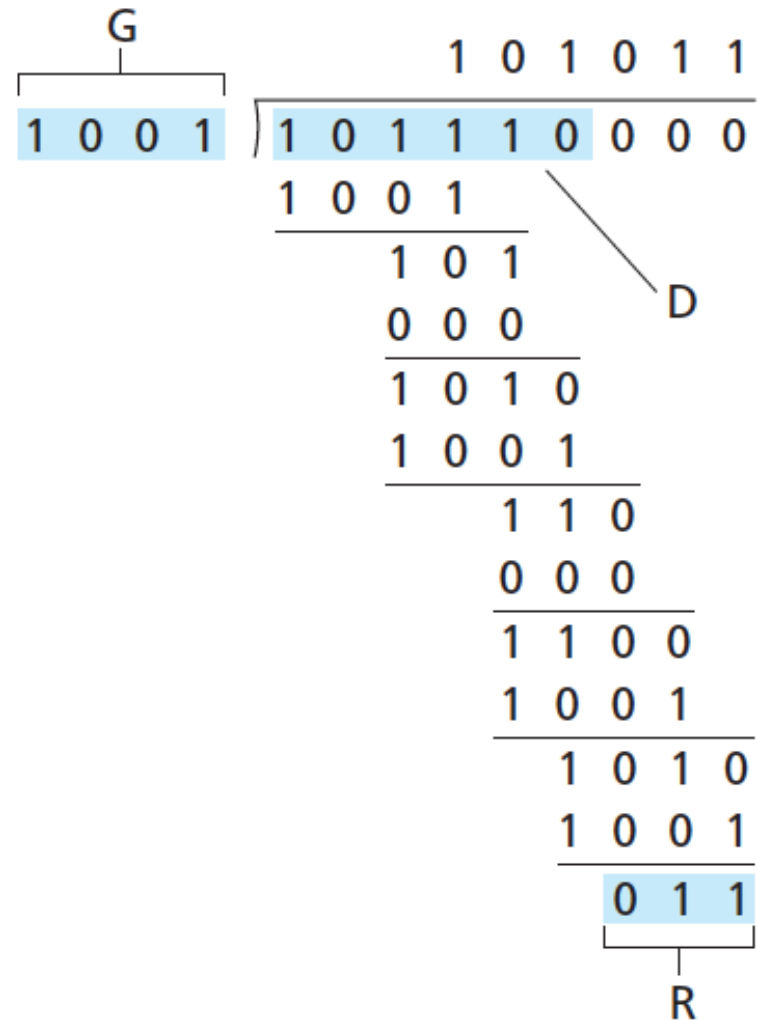
11

Link layer, LANs: outline

2.1 introduction, services

2.2 error detection, correction

2.3 multiple access protocols

2.4 LANs
- addressing, ARP
- Ethernet
- switches

Multiple access links, protocols
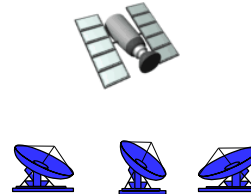
two types of "links":

- point-to-point
- *broadcast (shared wire or medium)*
  - old-fashioned Ethernet
  - 802.11 wireless LAN

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

- 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

- 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 protocol

given: broadcast channel of rate R bps

targets:

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
4. simple

MAC protocols: taxonomy

broad classes:

- *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

Channel partitioning MAC protocols: 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
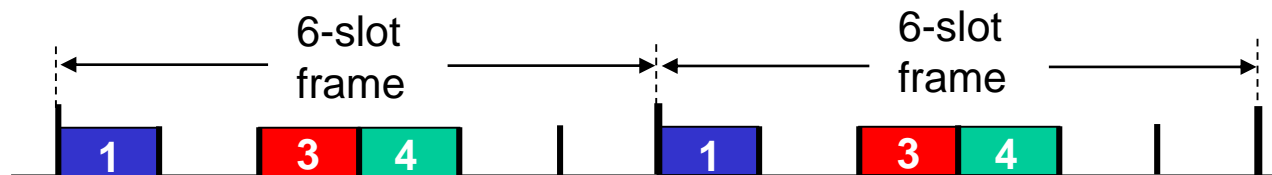
Channel partitioning MAC protocols: FDMA

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

FDM cable

time

frequency bands
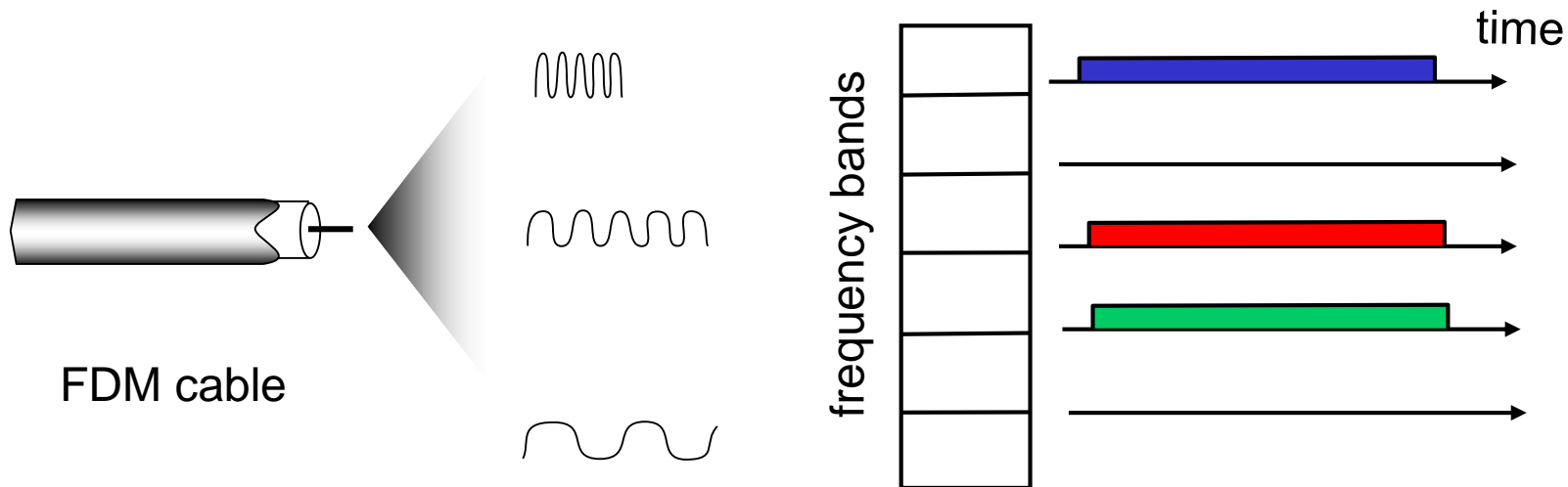
Random access protocols

- when node has packet to send
  - transmit at full channel data rate R.
  - 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
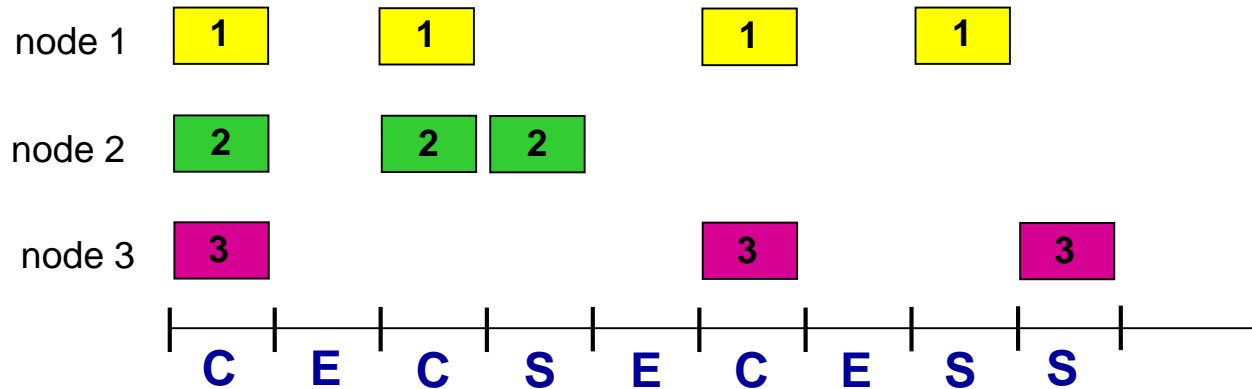 - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA

assumptions:

- all frames 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 with probability p

Slotted ALOHA

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 (unslotted) ALOHA

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

will overlap with start of ← i's frame → will overlap with end of ← i's frame →

node i frame

t_0-1 t_0 t_0+1
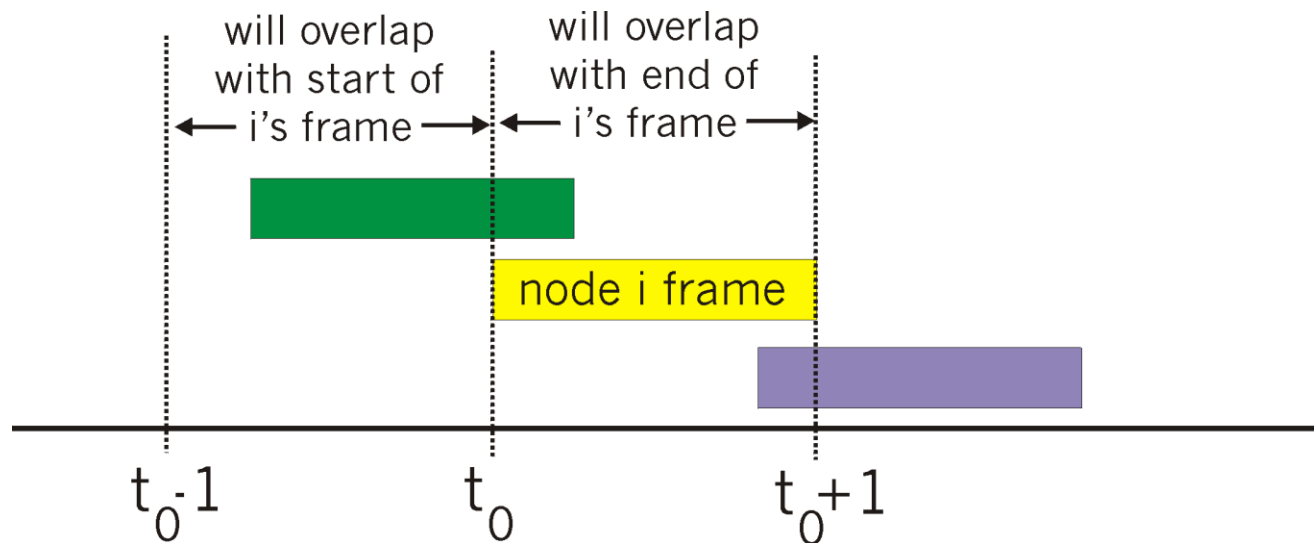
Pure ALOHA efficiency

P(success by given node) = P(node transmits) ·

P(no other node transmits in $[t_0-1, t_0]$ ·
P(no other node transmits in $[t_0-1, t_0]$

$$= p \cdot (1-p)^{N-1} \cdot (1-p)^{N-1}$$
$$= p \cdot (1-p)^{2(N-1)}$$

… choosing optimum p and then letting $n \longrightarrow \infty$

$$= 1/(2e) = .18$$

even *worse* than slotted Aloha!

CSMA (carrier sense multiple access)

CSMA: listen before transmit:

if channel sensed idle: transmit entire frame

- if channel sensed busy, defer transmission

- human analogy: don't interrupt others!

CSMA collisions

spatial layout of nodes

- **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 determining collision probability
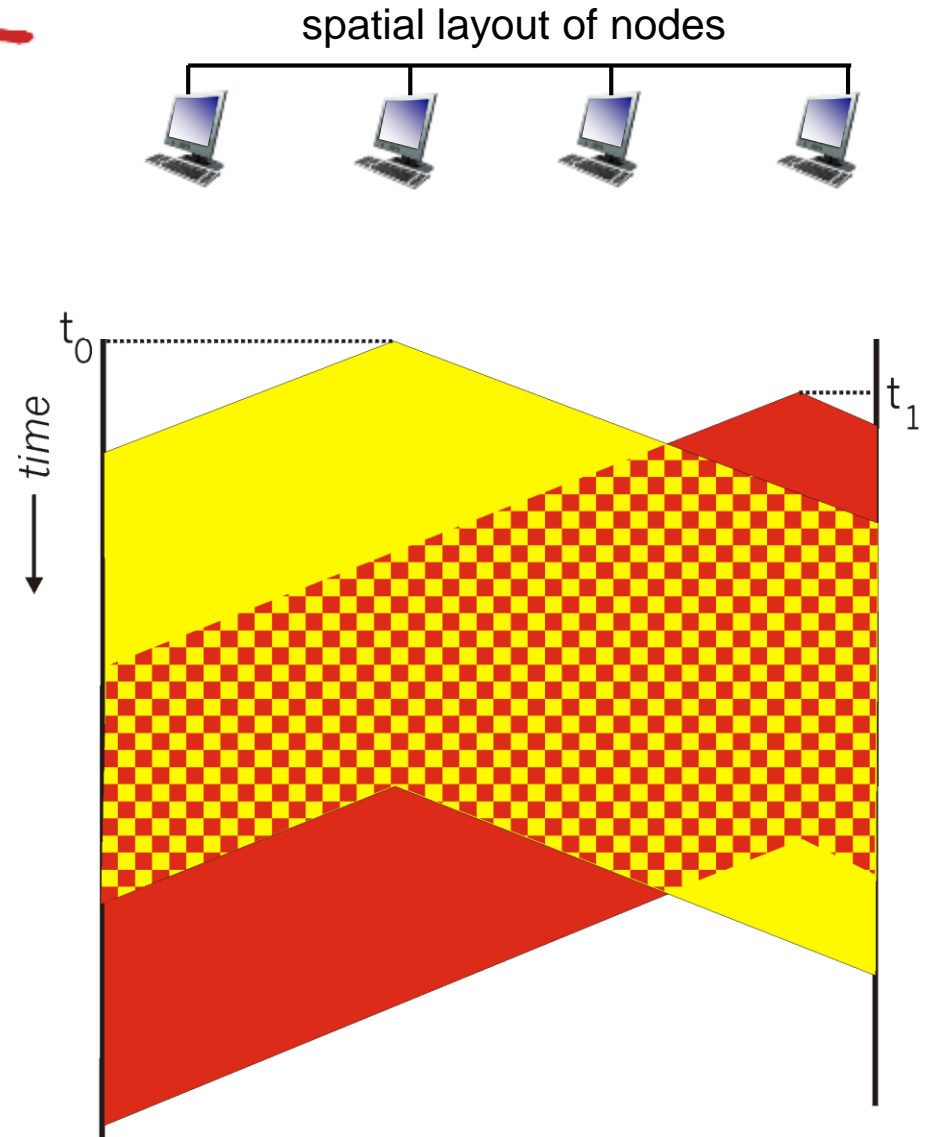
CSMA/CD (collision detection)

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

CSMA/CD (collision detection)

spatial layout of nodes

time

t_0

t_1

collision
detect/abort
time
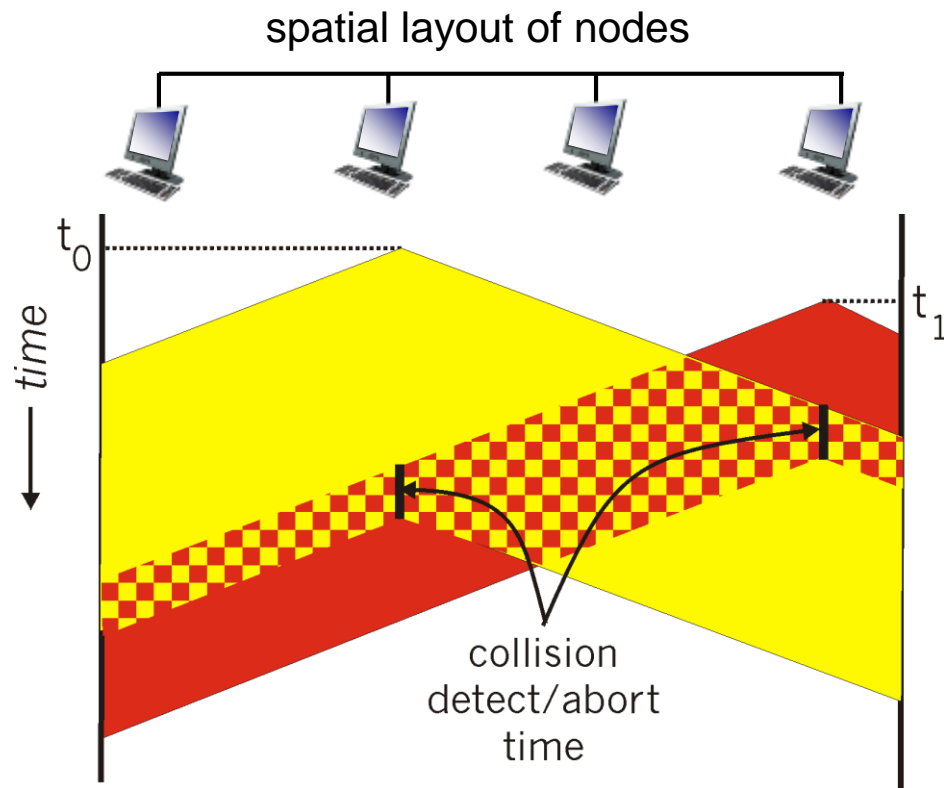
Ethernet CSMA/CD algorithm

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 (exponential) backoff:*

   - after mth collision, NIC chooses K at random from $\{0,1,2, \ldots, 2^m\text{-}1\}$. NIC waits $K\cdot512$ bit times, returns to Step 2

   - longer backoff interval with more collisions

CSMA/CD efficiency

- T_{prop} = max prop delay between 2 nodes in LAN
- t_{trans} = time to transmit max-size frame

$$efficiency = \frac{1}{1 + 5t_{prop}/t_{trans}}$$

- efficiency goes to 1
  - as t_{prop} goes to 0
  - as t_{trans} goes to infinity
- better performance than ALOHA: and simple, cheap, decentralized!

Summary of MAC protocols

- *channel partitioning,* by time, frequency or code
  - Time Division, Frequency Division

- *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

Link layer, LANs: outline

2.1 introduction, services

2.2 error detection, correction

2.3 multiple access protocols

2.4 LANs
- addressing, ARP
- Ethernet
- switches

MAC addresses and ARP

- **32-bit IP address:**
  - *network-layer* address for interface
  - used for layer 3 (network layer) forwarding

- **MAC (or LAN or physical or Ethernet) address:**
  - function: *used 'locally'' to get frame from one interface to another physically-connected interface*
  - 48 bit MAC address (for most LANs) burned in NIC ROM, also sometimes software settable
  - e.g.: 1A-2F-BB-76-09-AD

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

LAN addresses and ARP

each adapter on LAN has unique *LAN* address

1A-2F-BB-76-09-AD

LAN
(wired or
wireless)

71-65-F7-2B-08-53

58-23-D7-FA-20-B0

0C-C4-11-6F-E3-98

adapter
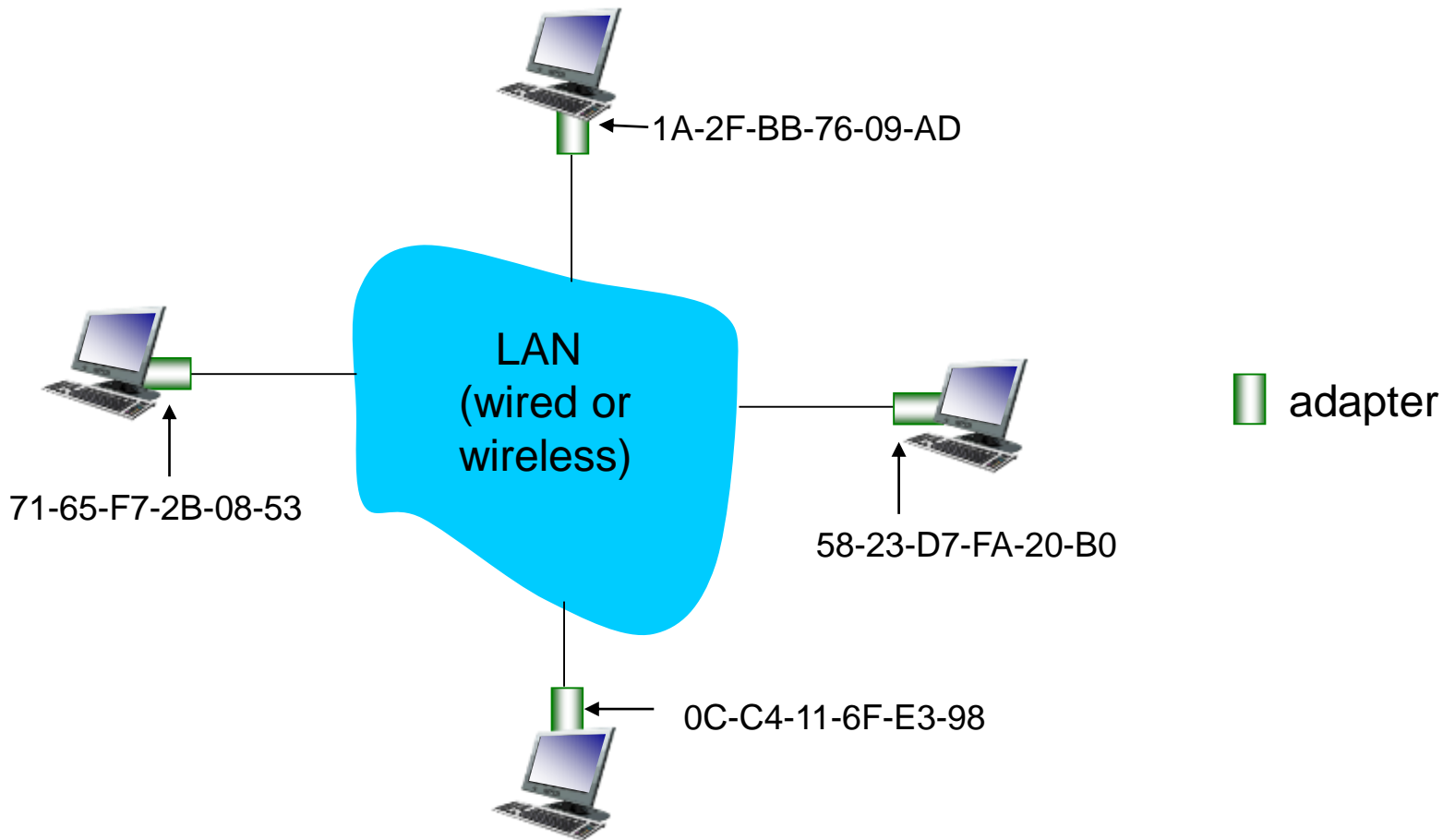
LAN addresses (more)

- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space (to assure uniqueness)
- MAC flat address ➔ portability
  - can move LAN card from one LAN to another
- IP hierarchical address *not* portable
  - address depends on IP subnet to which node is attached

ARP: address resolution protocol

Question: how to determine interface's MAC address, knowing its IP address?
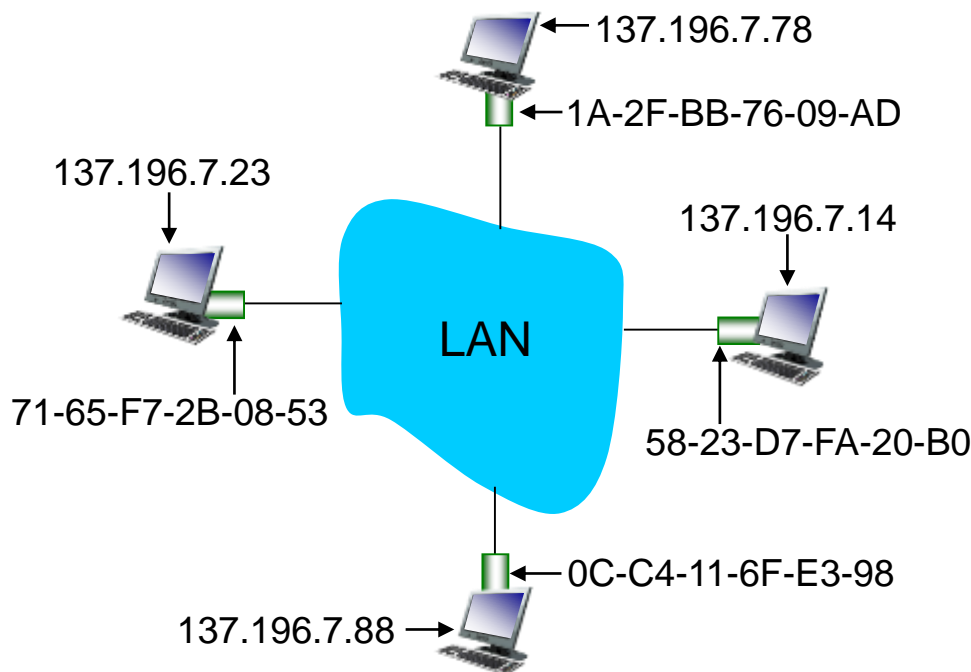
ARP table: each IP node (host, router) on LAN has table

- IP/MAC address mappings for some LAN nodes:

  < IP address; MAC address; TTL>

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

137.196.7.78

1A-2F-BB-76-09-AD

137.196.7.23

137.196.7.14

LAN

71-65-F7-2B-08-53

58-23-D7-FA-20-B0

0C-C4-11-6F-E3-98

137.196.7.88

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
  - destination 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)

Addressing: routing to another LAN

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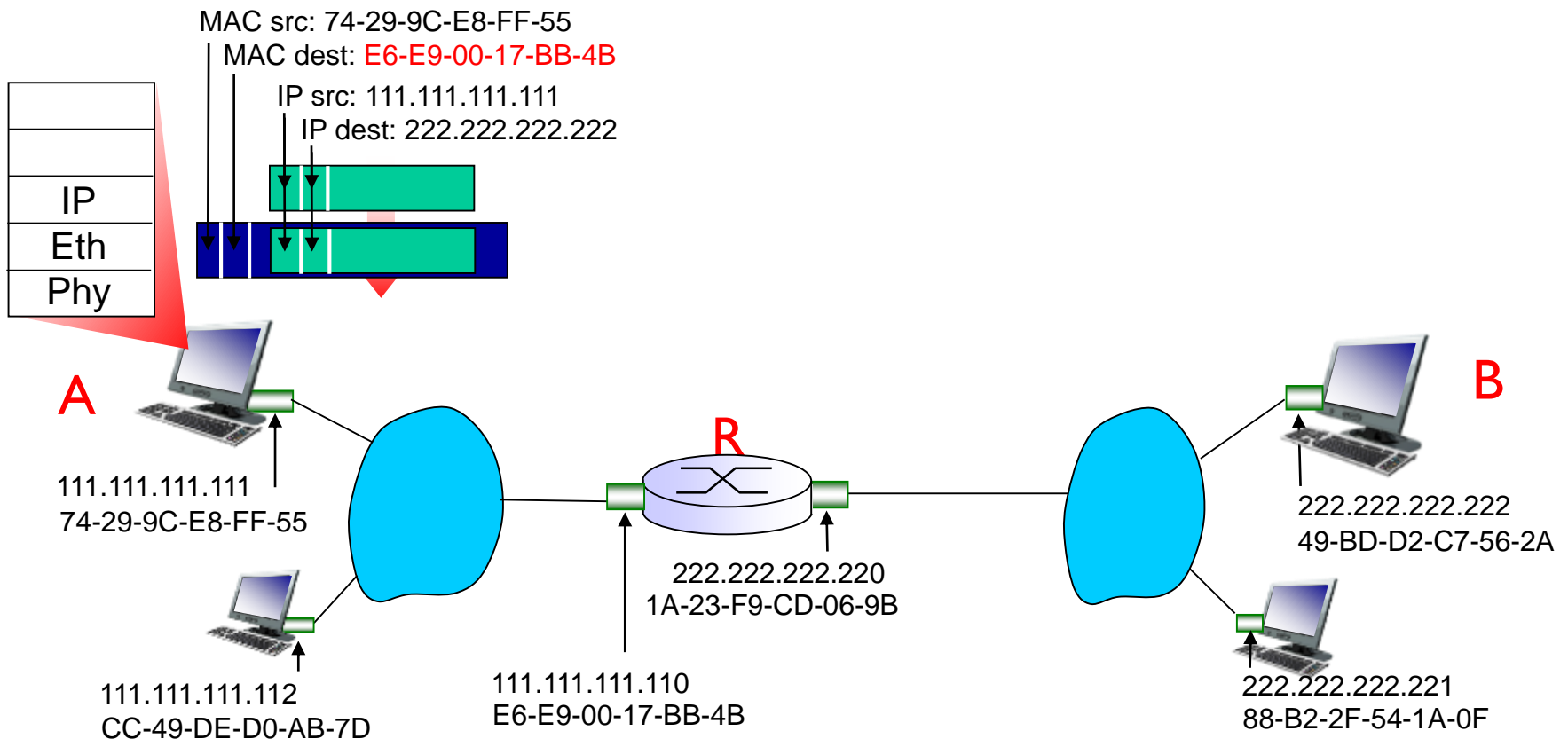
- focus on addressing – at IP (datagram) and MAC layer (frame)
- assume A knows B's IP address
- assume A knows IP address of first hop router, R
- MAC address?

A

111.111.111.111
74-29-9C-E8-FF-55

111.111.111.112
CC-49-DE-D0-AB-7D

R

111.111.111.110
E6-E9-00-17-BB-4B

222.222.222.220
1A-23-F9-CD-06-9B

B

222.222.222.222
49-BD-D2-C7-56-2A

222.222.222.221
88-B2-2F-54-1A-0F
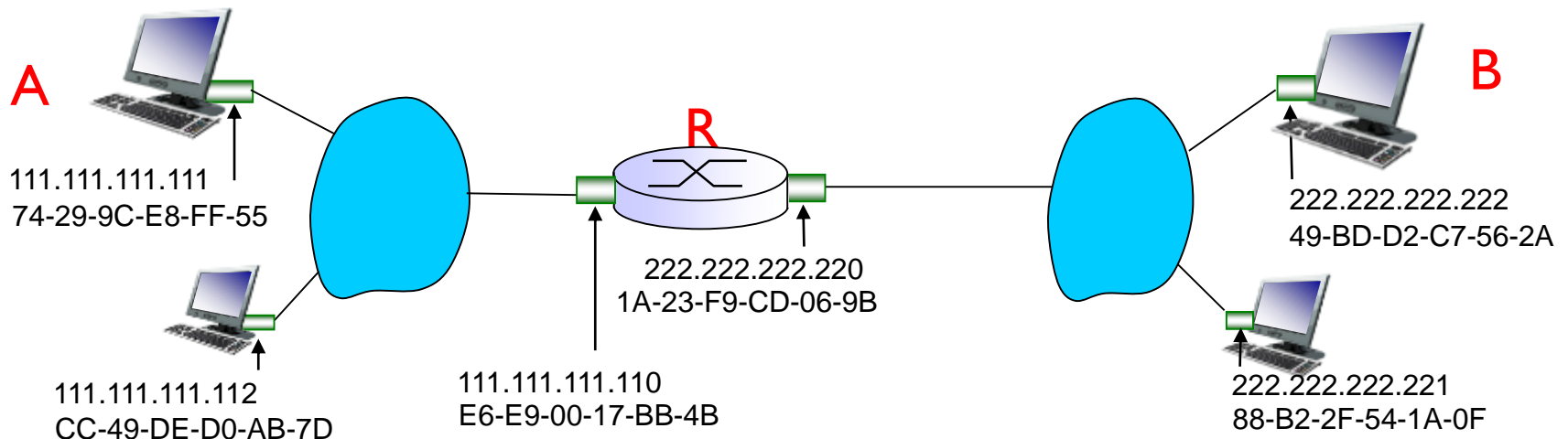
Addressing: routing to another LAN

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

MAC src: 74-29-9C-E8-FF-55
MAC dest: E6-E9-00-17-BB-4B

IP src: 111.111.111.111
IP dest: 222.222.222.222

IP
Eth
Phy

A

111.111.111.111
74-29-9C-E8-FF-55

111.111.111.112
CC-49-DE-D0-AB-7D

R

222.222.222.220
1A-23-F9-CD-06-9B

111.111.111.110
E6-E9-00-17-BB-4B

B

222.222.222.222
49-BD-D2-C7-56-2A

222.222.222.221
88-B2-2F-54-1A-0F

Addressing: routing to another LAN

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

MAC src: 74-29-9C-E8-FF-55
MAC dest: E6-E9-00-17-BB-4B
IP src: 111.111.111.111
IP dest: 222.222.222.222

IP
Eth
Phy

IP
Eth
Phy

A

B

111.111.111.111
74-29-9C-E8-FF-55

111.111.111.112
CC-49-DE-D0-AB-7D

222.222.222.220
1A-23-F9-CD-06-9B

111.111.111.110
E6-E9-00-17-BB-4B

222.222.222.222
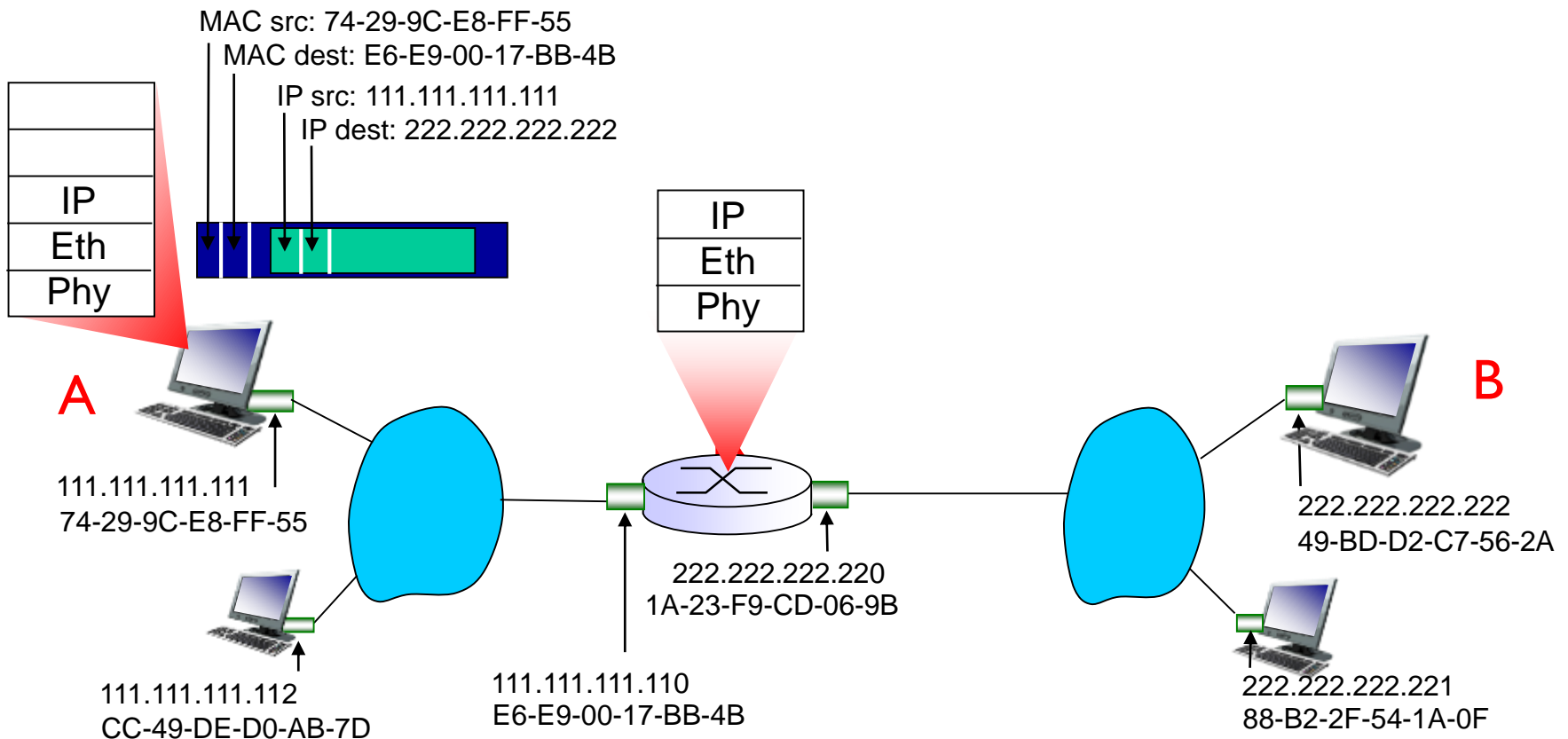49-BD-D2-C7-56-2A

222.222.222.221
88-B2-2F-54-1A-0F

Addressing: routing to another LAN

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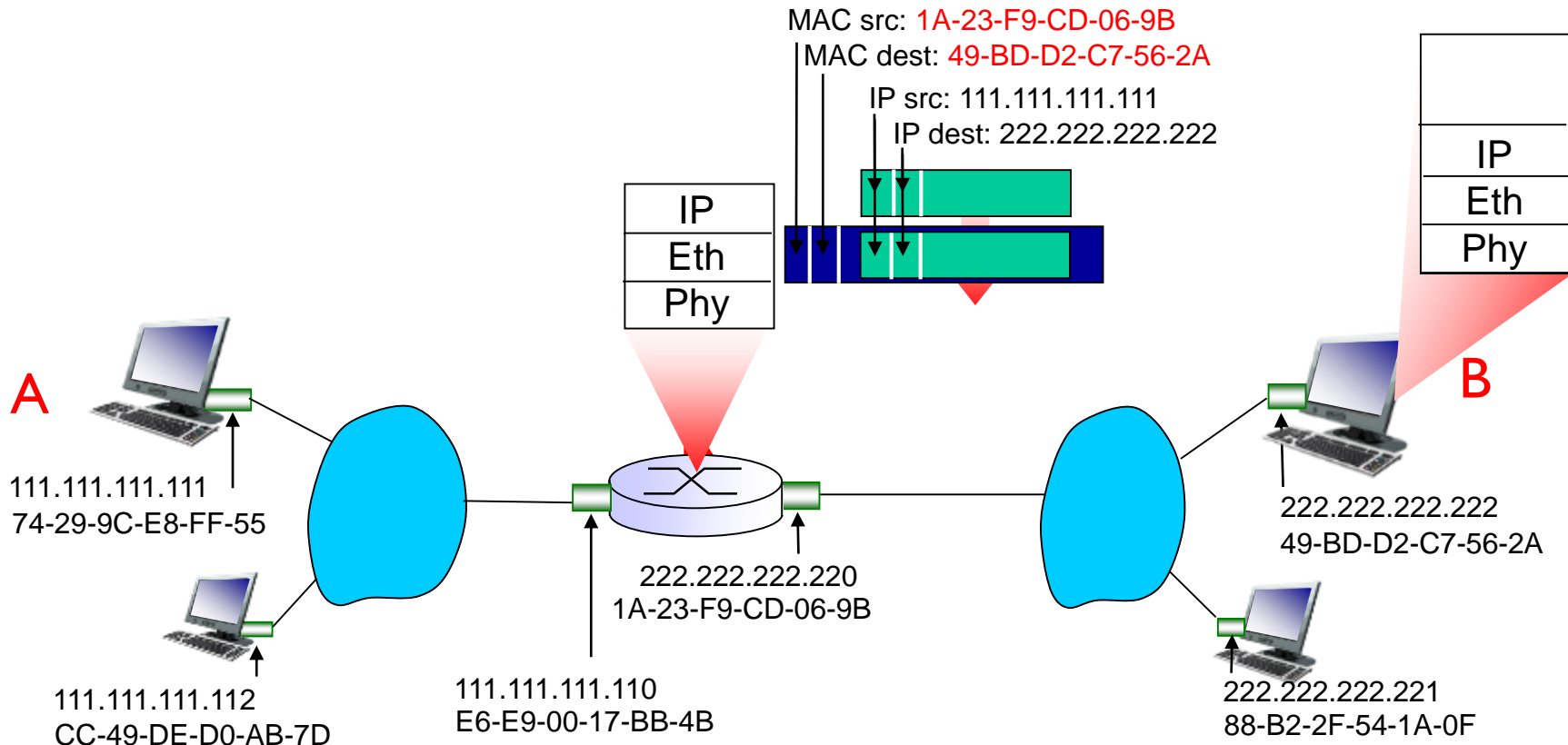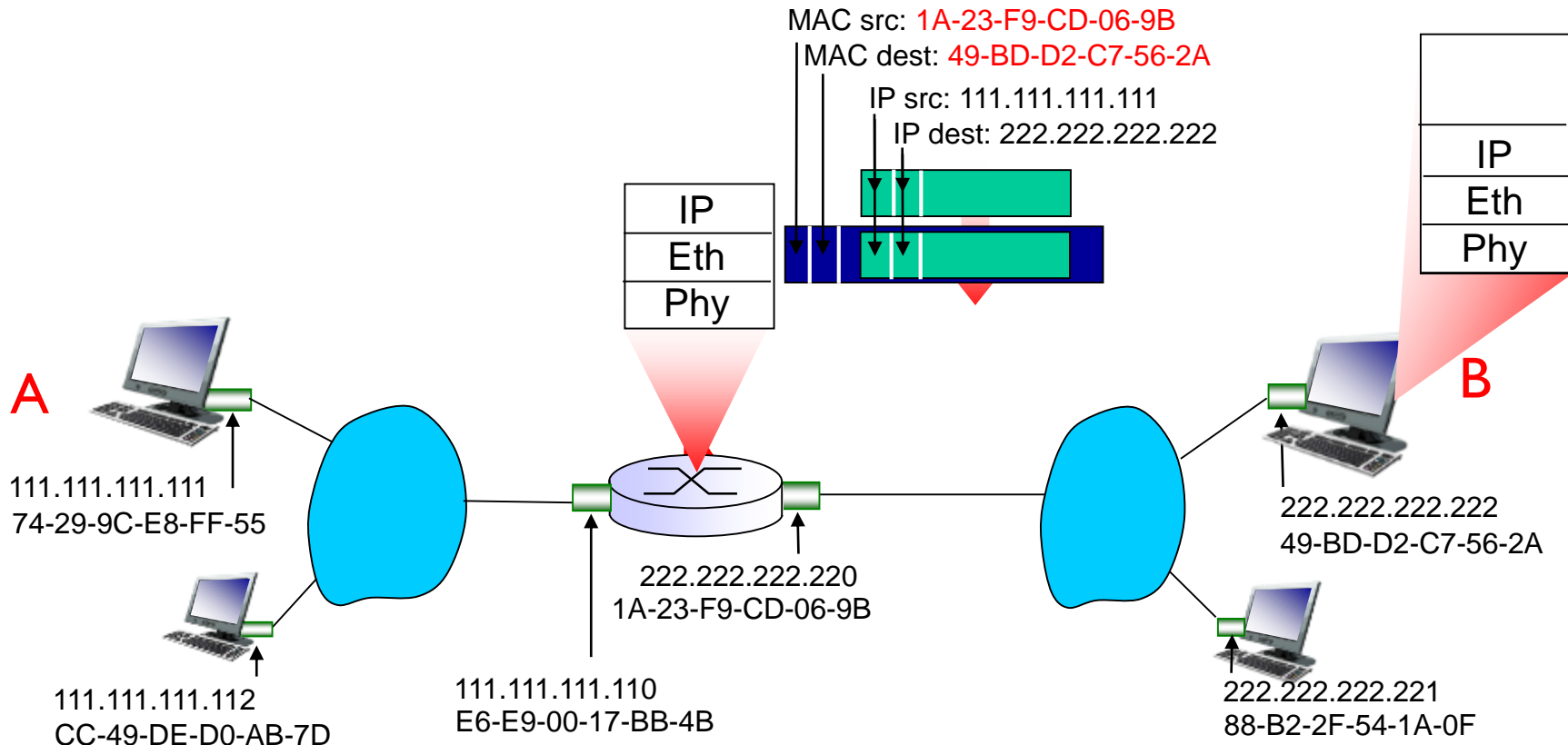- R creates link-layer frame with B's MAC address as destination address, frame contains A-to-B IP datagram

MAC src: 1A-23-F9-CD-06-9B
MAC dest: 49-BD-D2-C7-56-2A
IP src: 111.111.111.111
IP dest: 222.222.222.222

IP
Eth
Phy

IP
Eth
Phy

A

B

111.111.111.111
74-29-9C-E8-FF-55

111.111.111.112
CC-49-DE-D0-AB-7D

222.222.222.220
1A-23-F9-CD-06-9B

111.111.111.110
E6-E9-00-17-BB-4B

222.222.222.222
49-BD-D2-C7-56-2A
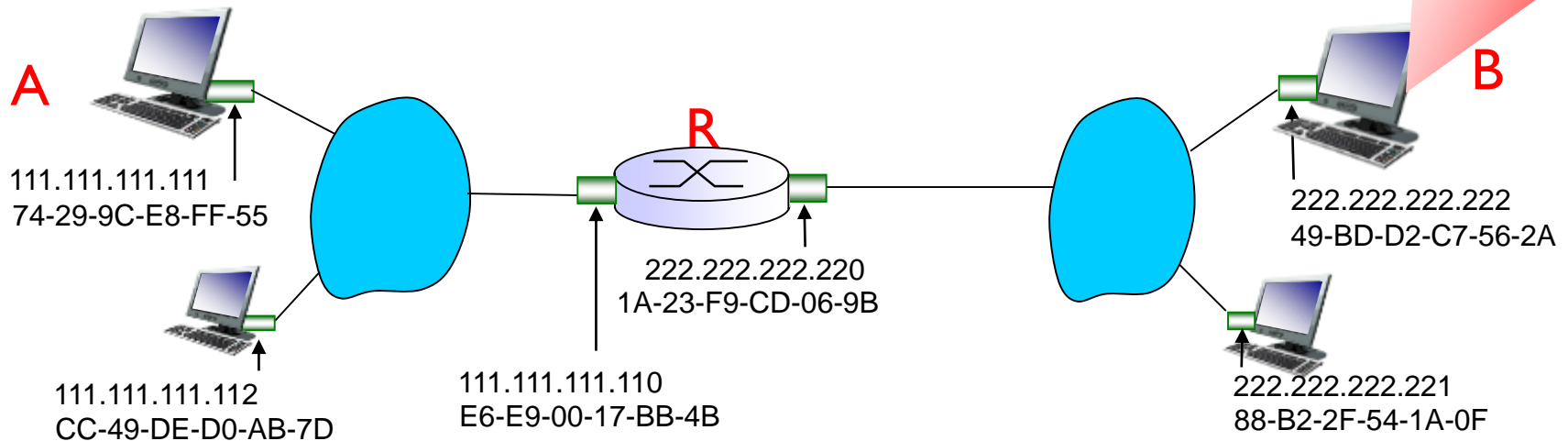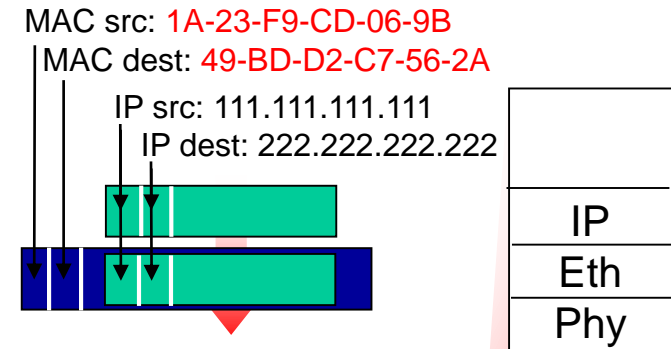
222.222.222.221
88-B2-2F-54-1A-0F

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 destination address, frame contains A-to-B IP datagram

MAC src: 1A-23-F9-CD-06-9B
MAC dest: 49-BD-D2-C7-56-2A
IP src: 111.111.111.111
IP dest: 222.222.222.222

IP
Eth
Phy

IP
Eth
Phy

A

B

111.111.111.111
74-29-9C-E8-FF-55

111.111.111.112
CC-49-DE-D0-AB-7D

222.222.222.220
1A-23-F9-CD-06-9B

111.111.111.110
E6-E9-00-17-BB-4B

222.222.222.222
49-BD-D2-C7-56-2A

222.222.222.221
88-B2-2F-54-1A-0F

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

MAC src: 1A-23-F9-CD-06-9B
MAC dest: 49-BD-D2-C7-56-2A
IP src: 111.111.111.111
IP dest: 222.222.222.222

IP
Eth
Phy

A
111.111.111.111
74-29-9C-E8-FF-55

R

222.222.222.220
1A-23-F9-CD-06-9B

B
222.222.222.222
49-BD-D2-C7-56-2A

111.111.111.112
CC-49-DE-D0-AB-7D

111.111.111.110
E6-E9-00-17-BB-4B

222.222.222.221
88-B2-2F-54-1A-0F

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

Link layer, LANs: outline

2.1 introduction, services
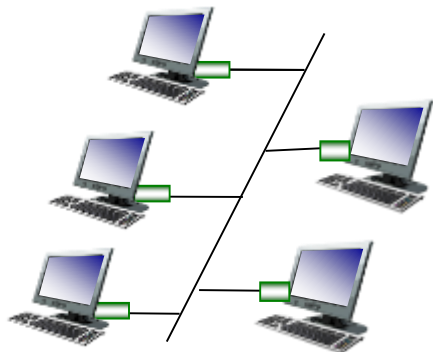
2.2 error detection, correction
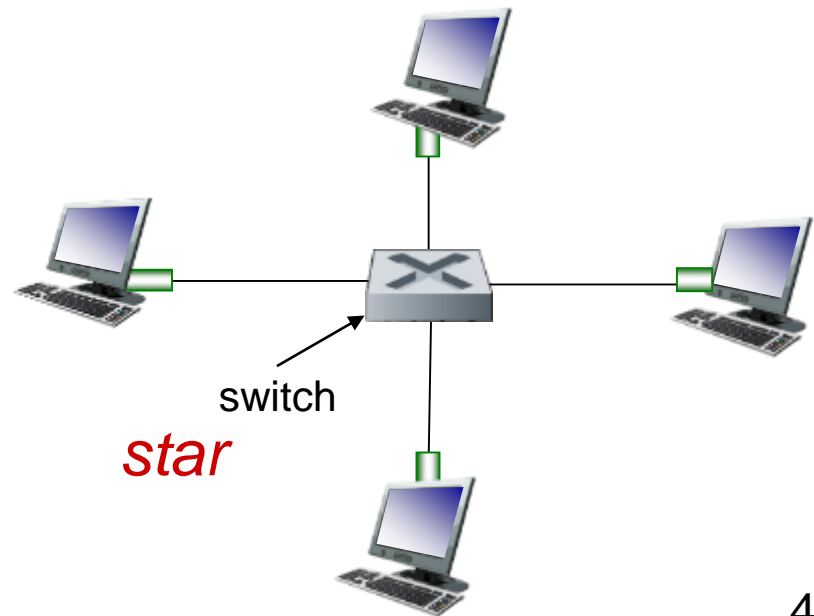
2.3 multiple access protocols

2.4 LANs
- addressing, ARP
- Ethernet
- switches

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

switch

star

Ethernet frame structure

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

preamble	dest. address	source address	*type*	data (payload)	CRC

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)
- *CRC:* cyclic redundancy check at receiver
  - error detected: frame is dropped

type

preamble	dest. address	source address		data (payload)	CRC

Ethernet: unreliable, connectionless

- *connectionless:* no handshaking between sending and receiving NICs

- *unreliable:* receiving NIC doesn't send acks or nacks to sending NIC
  - data in dropped frames recovered only if initial sender uses higher layer rdt (e.g., TCP), otherwise dropped data lost

- Ethernet's MAC protocol: unslotted *CSMA/CD with binary backoff*

Link layer, LANs: outline

2.1 introduction, services

2.2 error detection, correction

2.3 multiple access protocols

2.4 LANs
- addressing, ARP
- Ethernet
- switches

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: *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
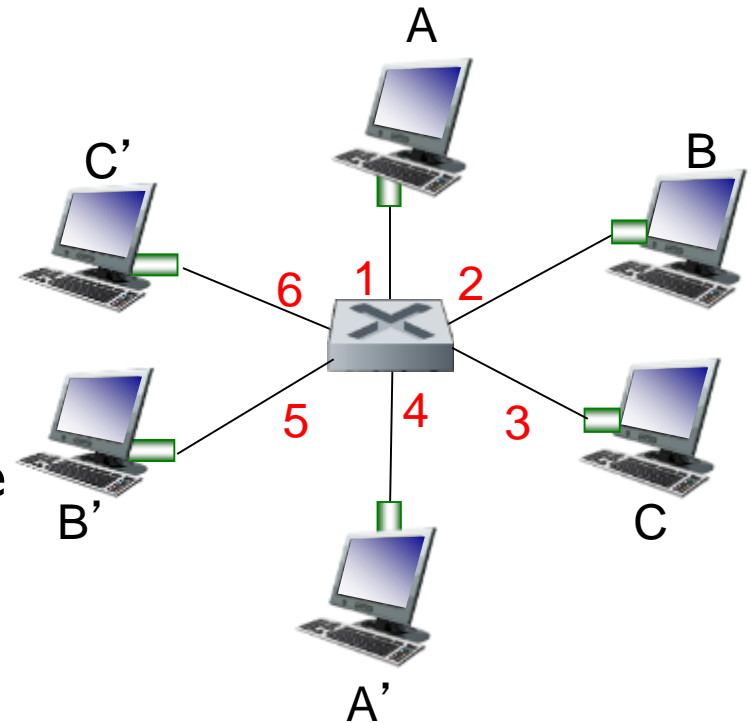- *switching:* A-to-A' and B-to-B' can transmit simultaneously, without collisions
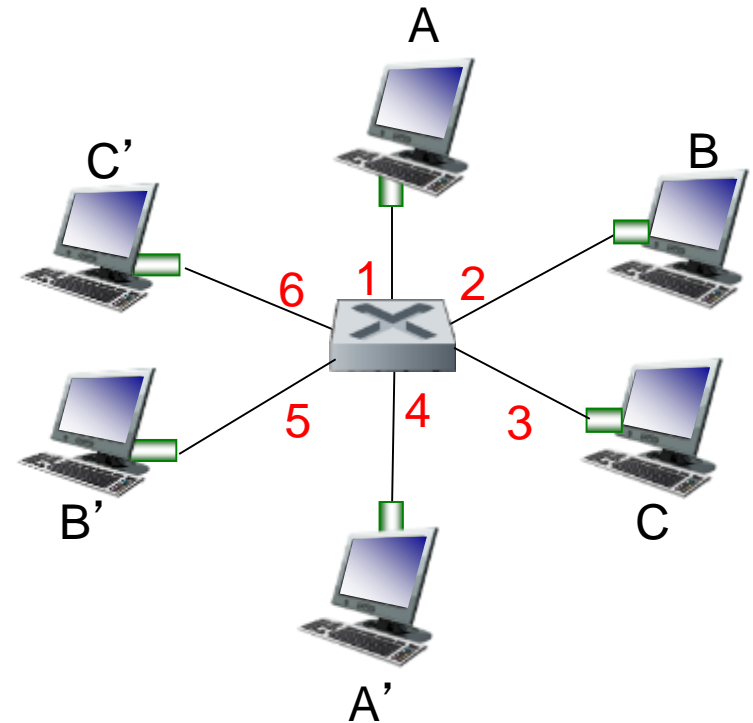
switch with six interfaces (1,2,3,4,5,6)

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)

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

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

52

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

Source: A
Dest: A'
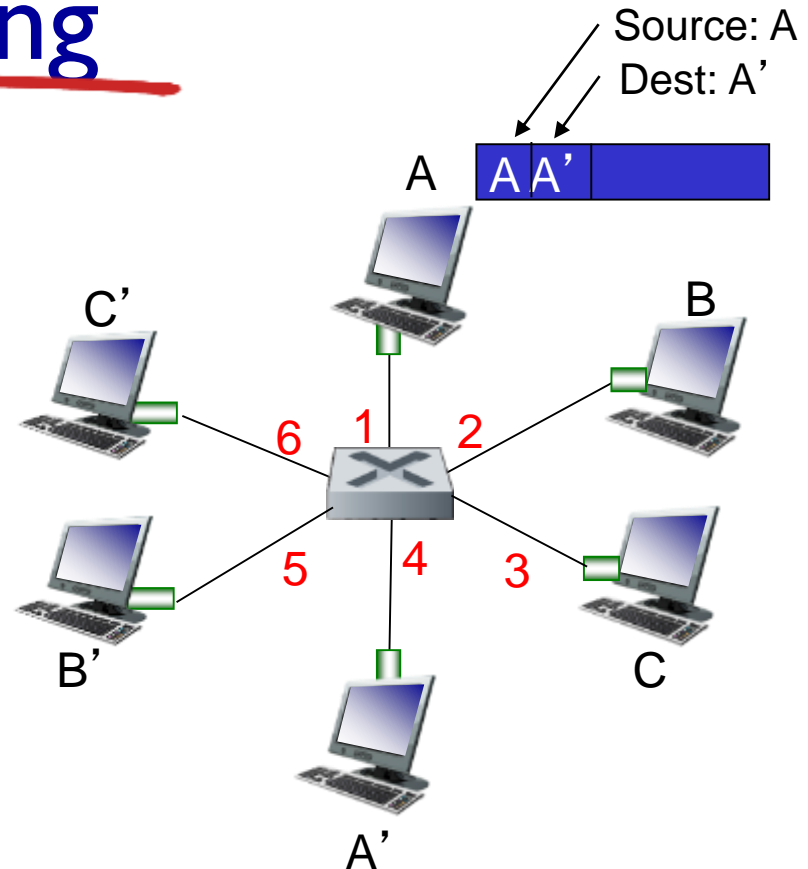
A | A A' |

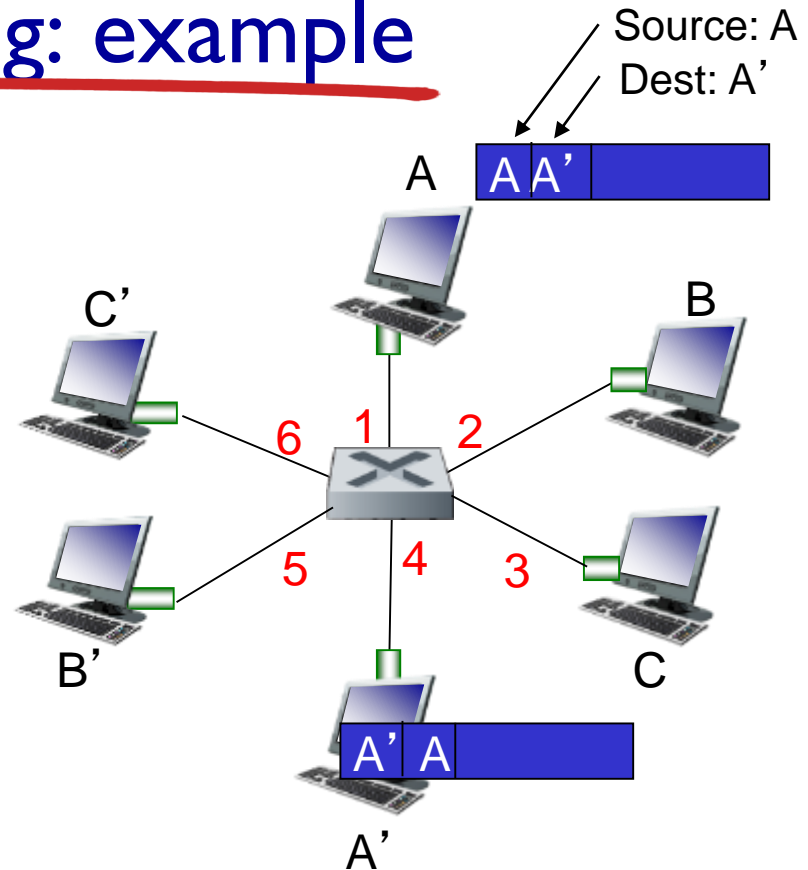| MAC addr	interface	TTL
A	*1*	*60*

Switch table (initially empty)

Switch: frame filtering/forwarding

when frame received at switch:

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

Self-learning, forwarding: example

- frame destination, A', location unknown: *flood*

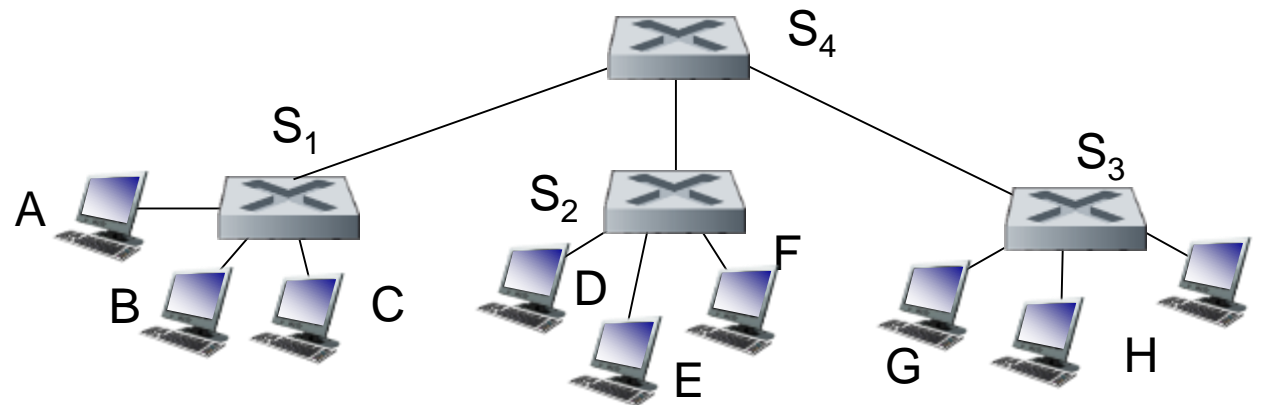- destination A location known: selectively send on just one link

Source: A
Dest: A'

MAC addr	interface	TTL
A	1	60
A'	4	60

switch table (initially empty)

Interconnecting switches

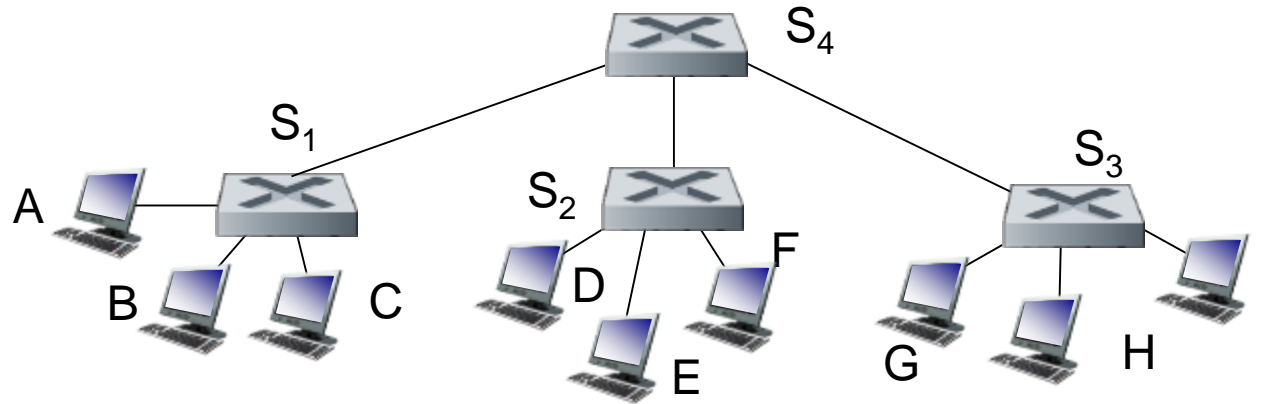self-learning switches can be connected together:

Q: sending from A to G - how does S_1 know to forward frame destined to G via S_4 and S_3?

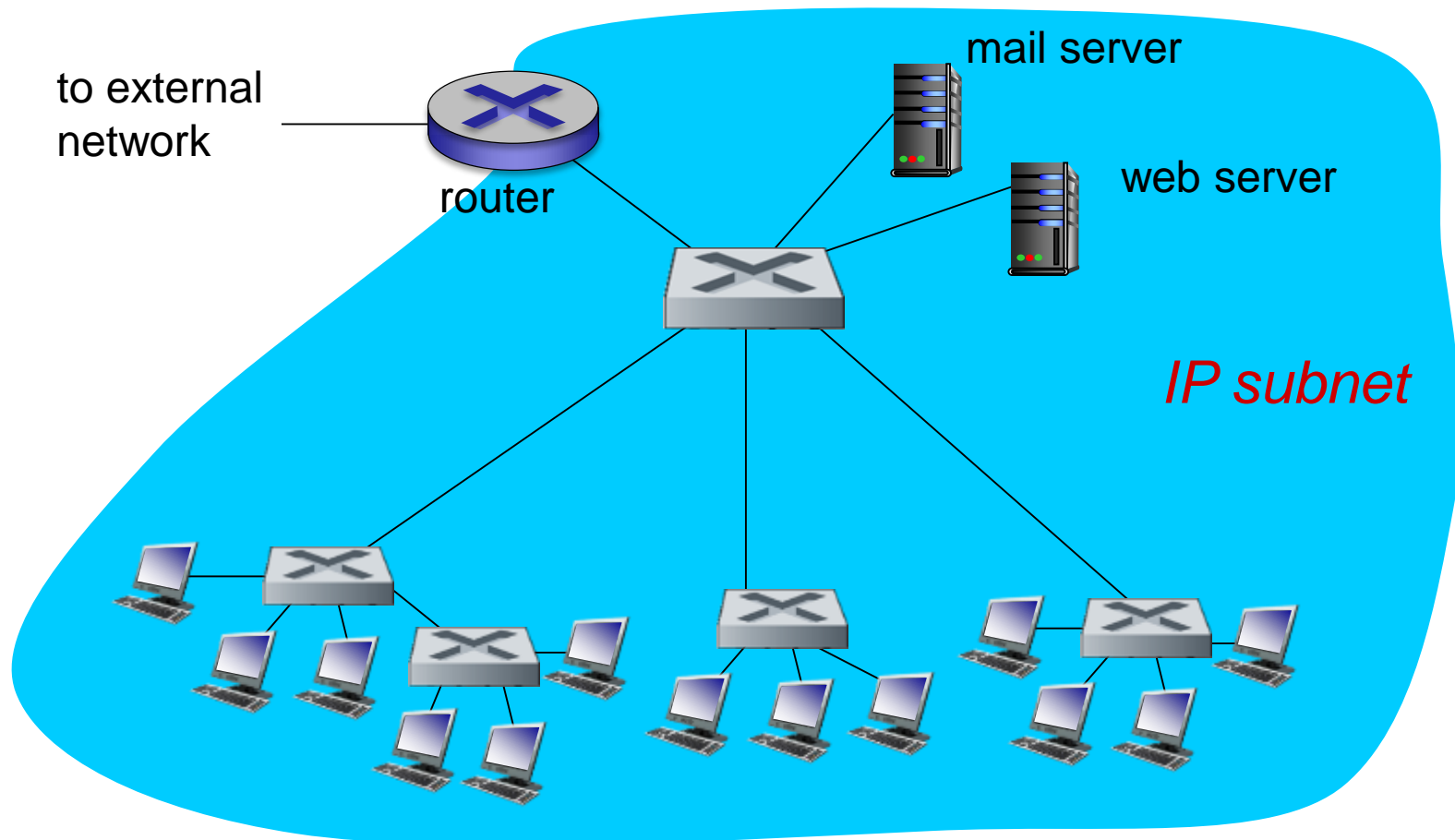- *A:* self learning! (works exactly the same as in single-switch case!)

Self-learning multi-switch example

Suppose C sends frame to I, I responds to C

- <u>Q:</u> show switch tables and packet forwarding in S_1, S_2, S_3, S_4

Institutional network

to external network

router

mail server

web server

IP subnet

Chapter 6: 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