TDTS06: Computer Networks

Instructor: Niklas Carlsson

Email: niklas.carlsson@liu.se

Notes derived from "*Computer Networking: A Top Down Approach*", by Jim Kurose and Keith Ross, Addison-Wesley.

The slides are adapted and modified based on slides from the book's companion Web site, as well as modified slides by Anirban Mahanti and Carey Williamson.

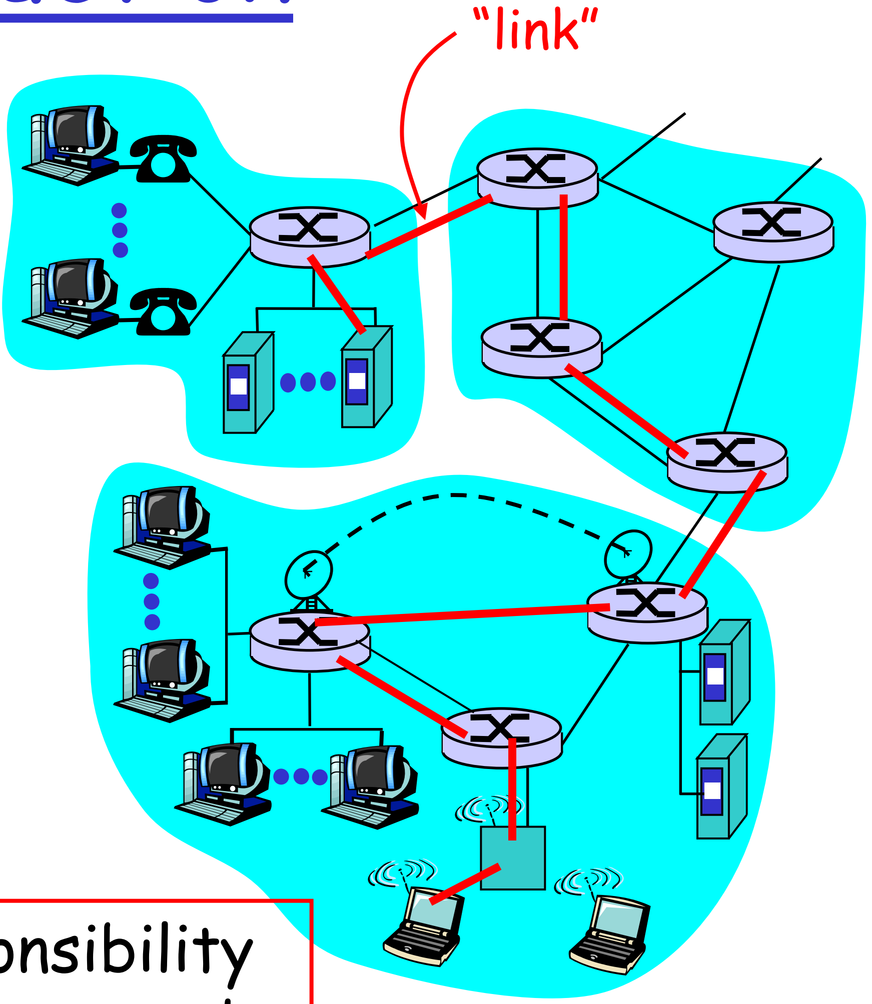
Our Goals

- understand principles behind data link layer services:
  - link-layer addressing
  - reliable data transfer, flow control
  - error detection and correction
  - sharing a broadcast channel: multiple access
- instantiation and implementation of various link layer technologies

Link Layer: Introduction

Some terminology:

- hosts and routers are **nodes**
- communication channels that connect adjacent nodes along communication path are **links**
  - wired links
  - wireless links
  - LANs
- layer-2 packet is a **frame**, encapsulates datagram



data-link layer (DLL) has responsibility of transferring datagram from one node to 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 wireless on last link
- Each DLL protocol provides different services
  - e.g., may or may not provide reliable data transfer over link

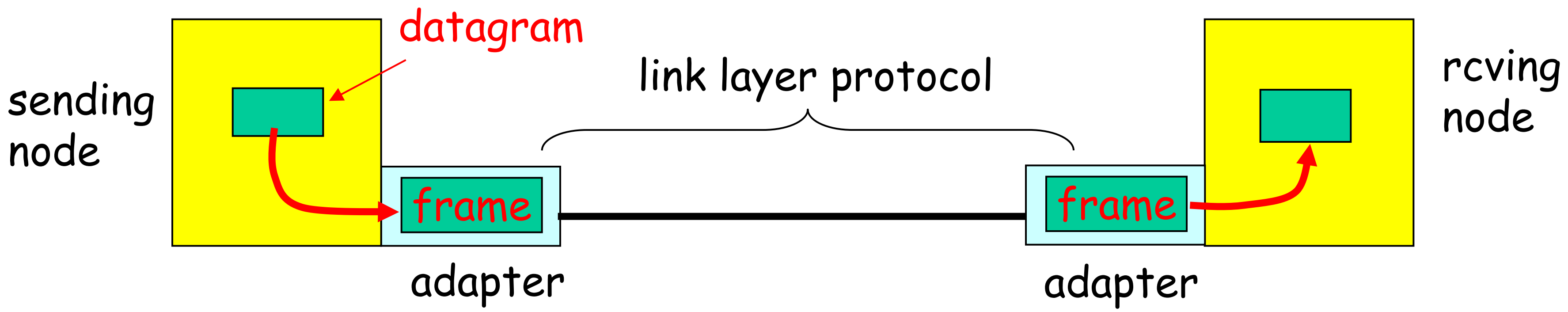
Link Layer Services

- Framing, link access:
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - "MAC" addresses used in frame headers to identify source, dest
    - different from IP address!
- Reliable delivery between adjacent nodes
  - we learned how to do this already (chapter 3)!
  - 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)

- Flow Control:
  - regulate transmissions between sender and receiver
- 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

Adaptors Communicating



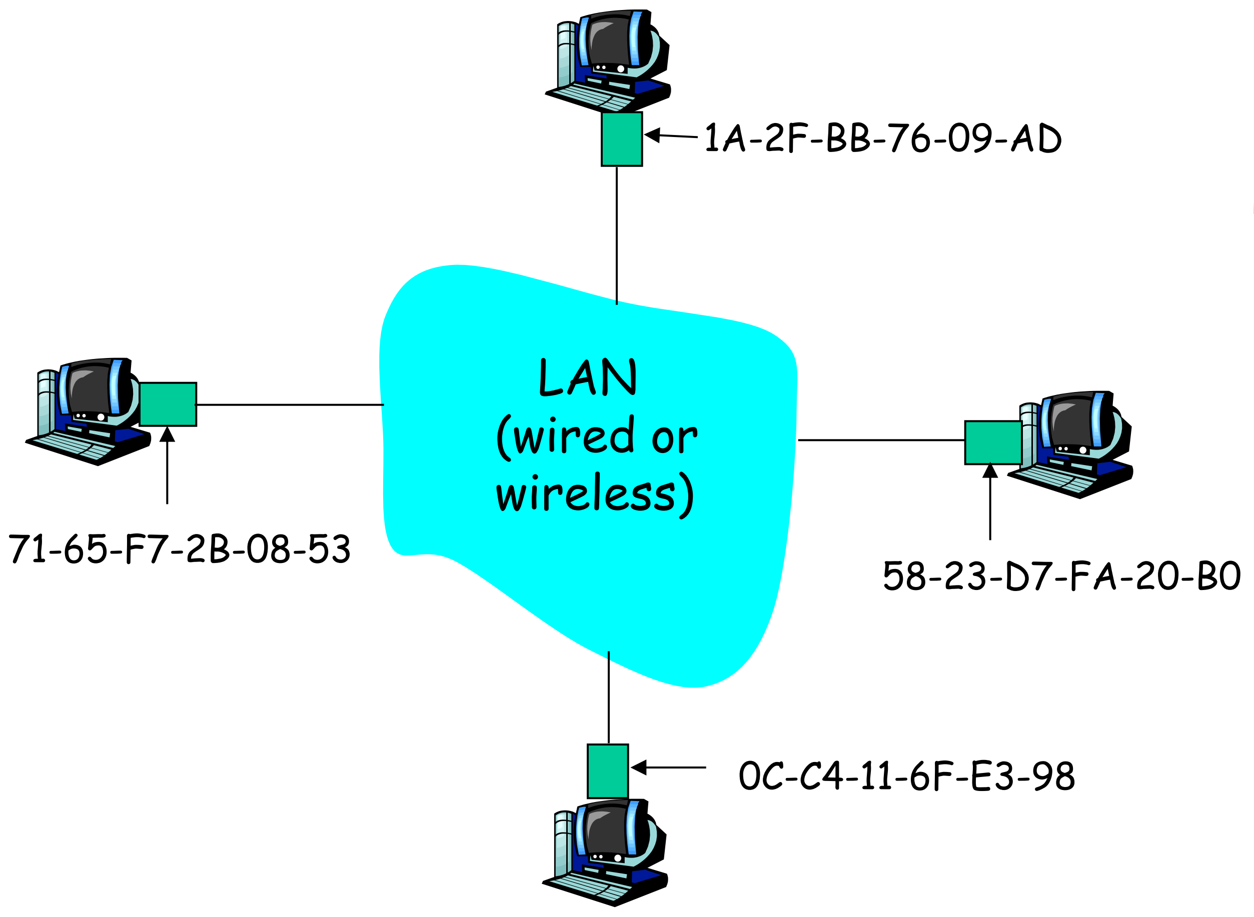
- link layer implemented in "adaptor" (aka NIC)
  - Ethernet card, PCMCI card, 802.11 card
- sending side:
  - encapsulates datagram in a frame
  - adds error checking bits, rdt, flow control, etc.
- receiving side
  - looks for errors, rdt, flow control, etc
  - extracts datagram, passes to rcving node
- adapter is semi-autonomous
- link & physical layers

MAC Addresses (1/3)

- 32-bit IP address:
  - *network-layer* address
  - used to get datagram to destination IP subnet
- MAC address (e.g., Ethernet LAN):
  - used to get datagram from one interface to another physically-connected interface (on the same network)
  - 48-bit MAC address (for most LANs) burned in the adapter ROM (globally unique)

MAC Addresses(2/3)

Each adapter on LAN has unique LAN address



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

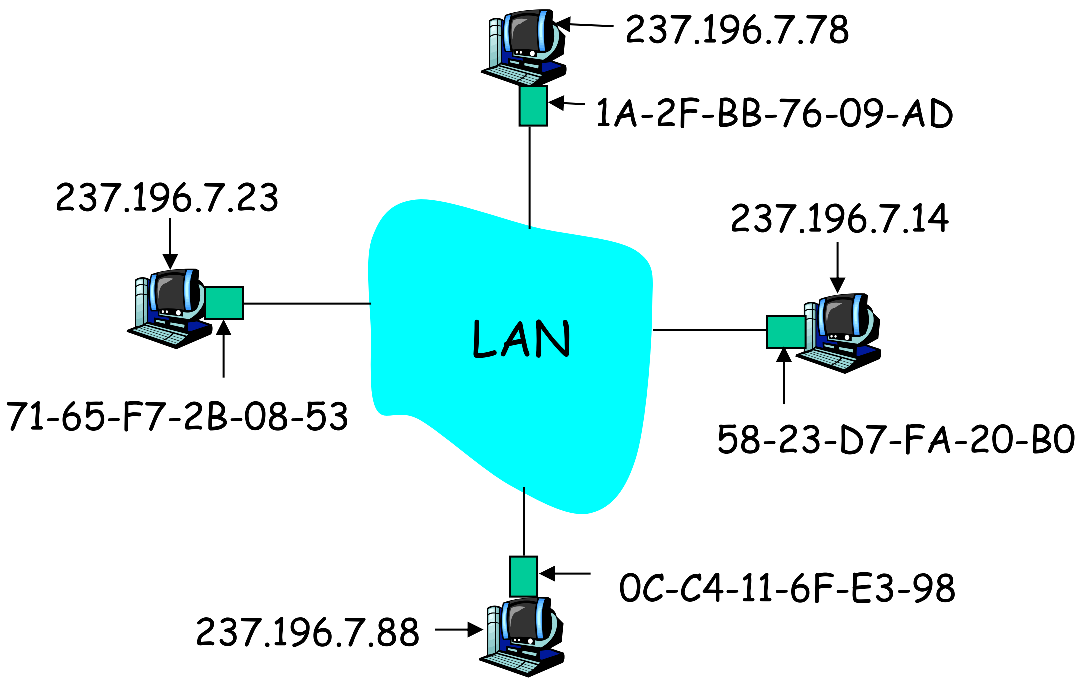
🟩 = adapter

LAN Address (3/3)

- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space
- MAC flat address provides portability
  - can move LAN card from one LAN to another
  - different than with IP addresses!

ARP: Address Resolution Protocol

Question: how to determine MAC address of B knowing B's IP address?



- 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)

ARP protocol: Same LAN (network)

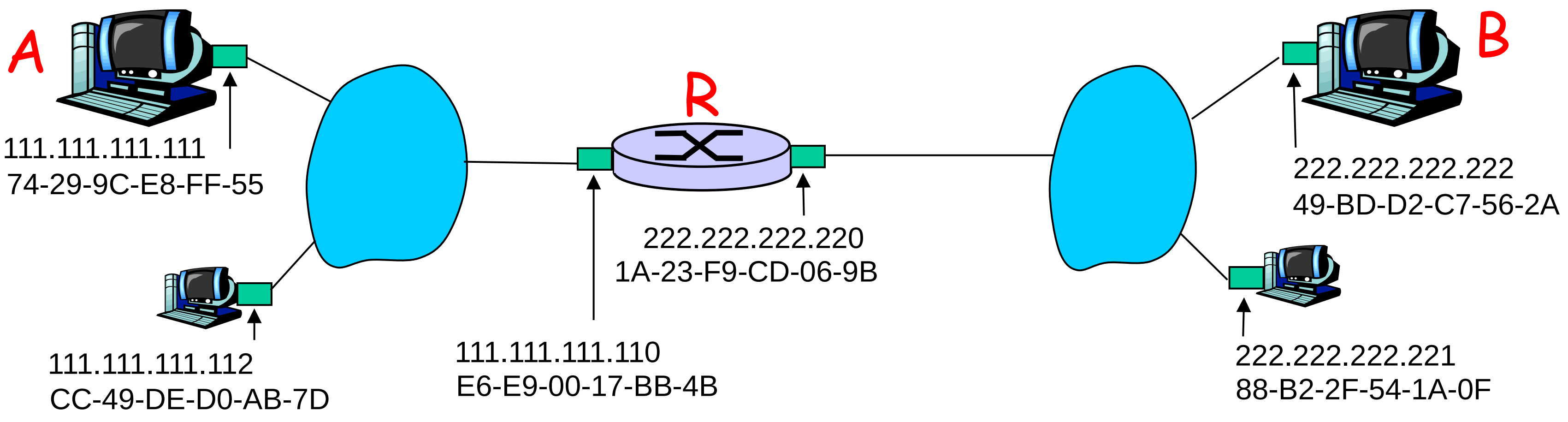
- A wants to send datagram to B, and B's MAC address not in A's ARP table.
- A broadcasts ARP query packet, containing B's IP address
  - Dest MAC address = FF-FF-FF-FF-FF-FF
  - all machines on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
  - frame sent to A's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

- A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
- ARP is a "soft state" protocol: information that times out unless refreshed
- ARP is "plug-and-play":
  - nodes create their ARP tables without intervention from net administrator

Addressing: routing to another LAN

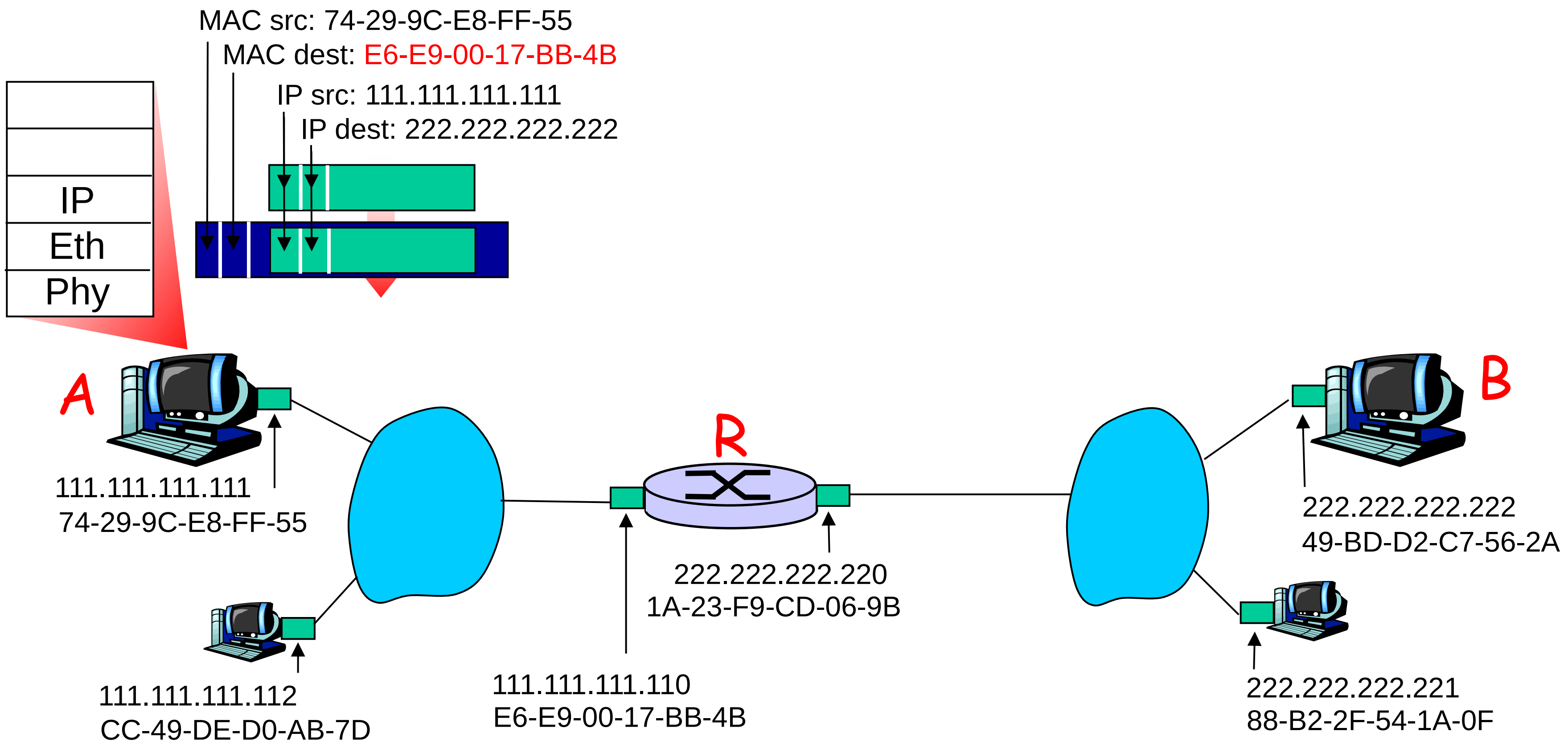
walkthrough: send datagram from A to B via R.

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



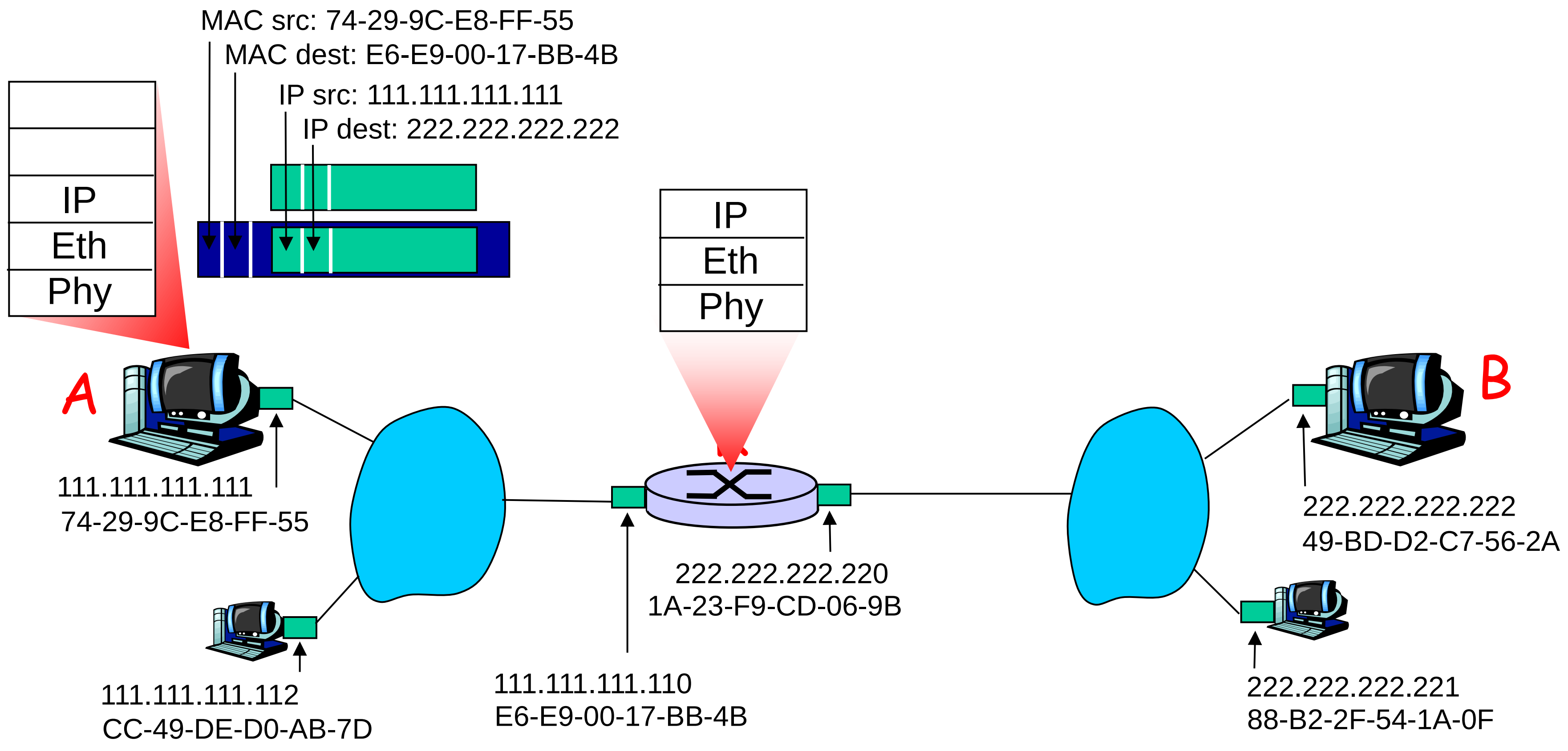
Addressing: routing to another LAN

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



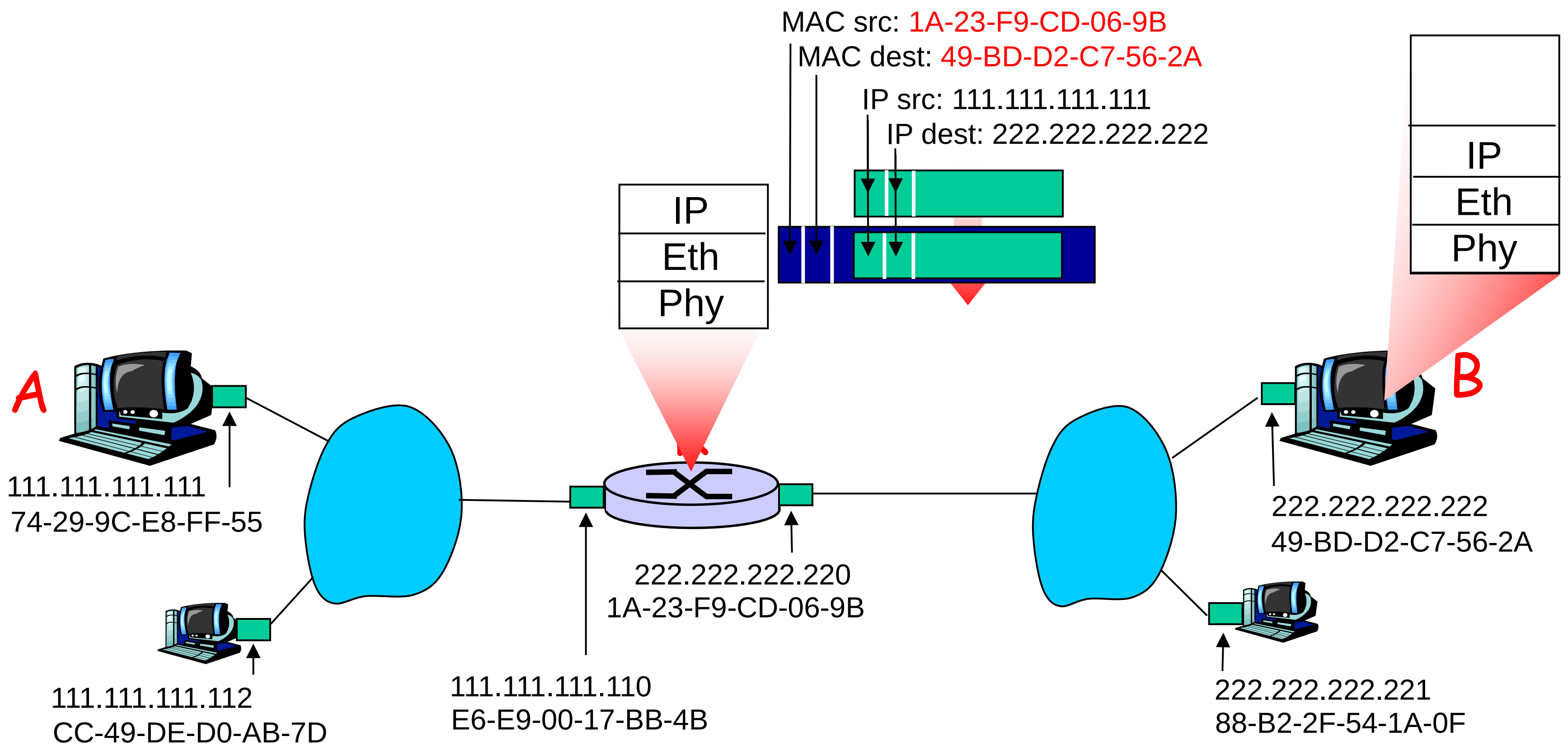
Addressing: routing to another LAN

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



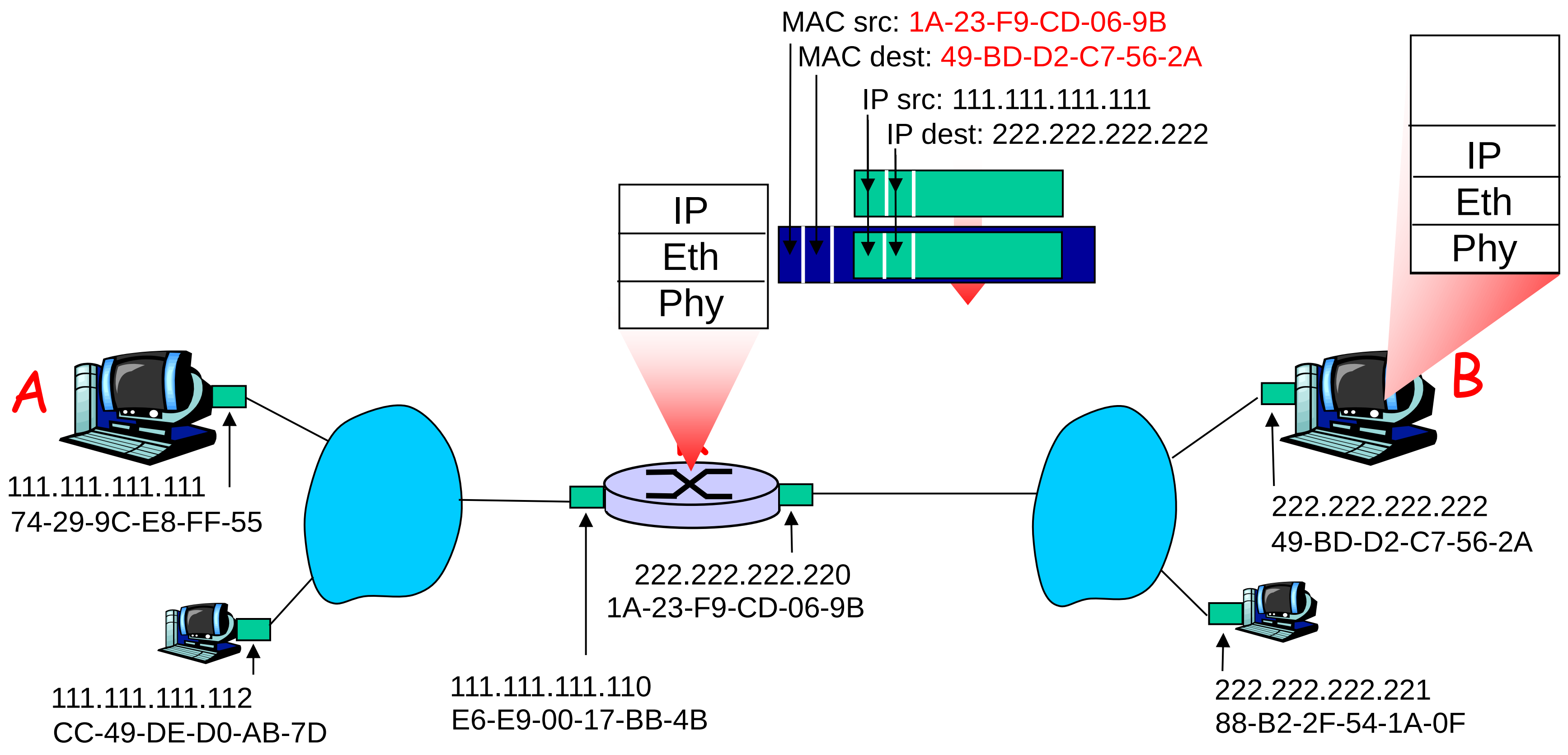
Addressing: routing to another LAN

- 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



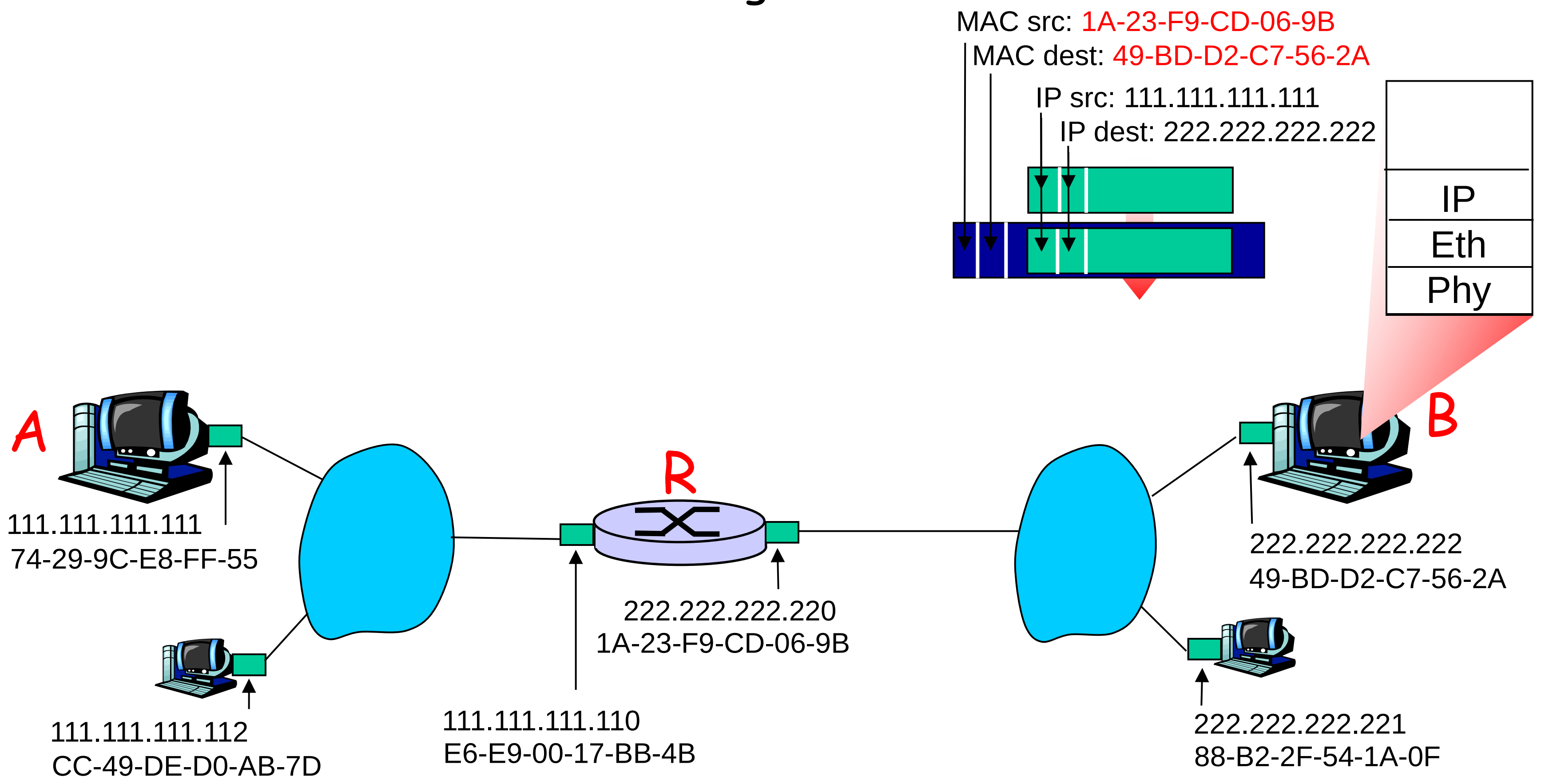
Addressing: routing to another LAN

- 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



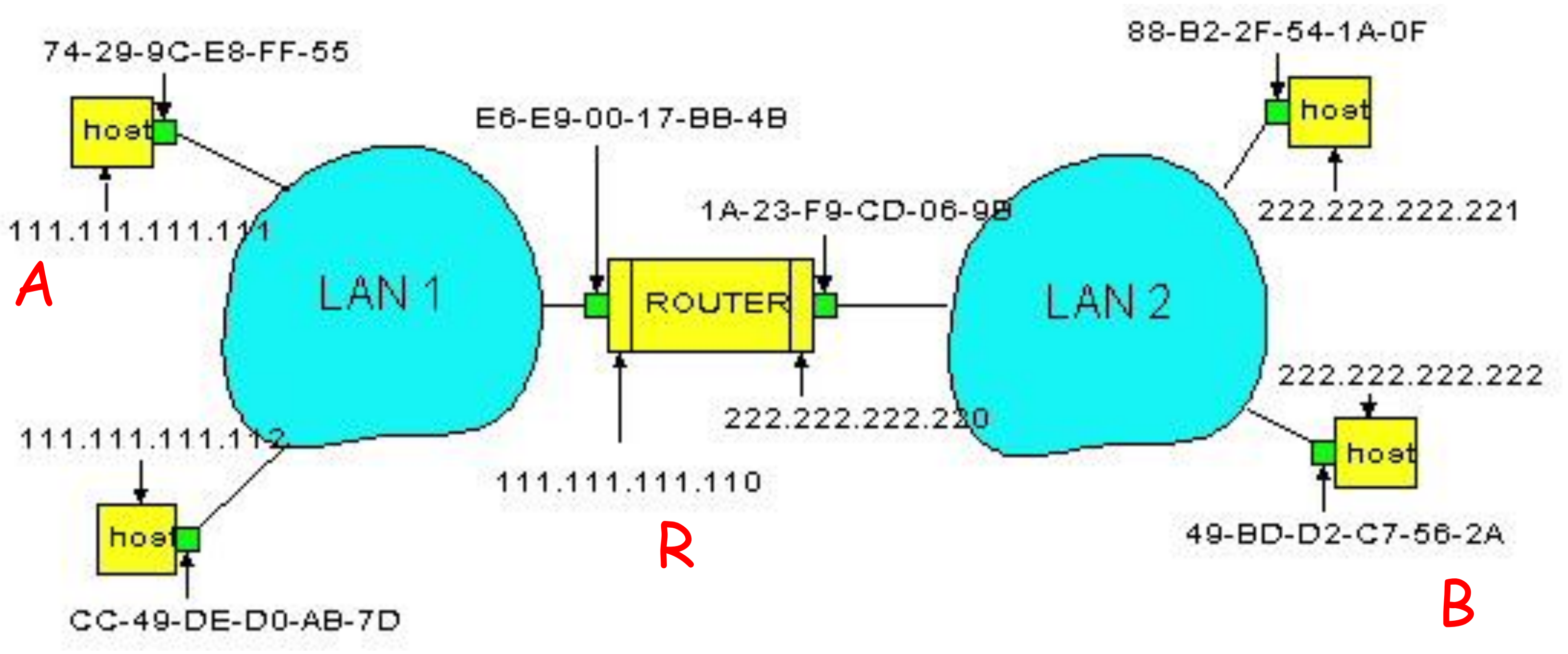
Addressing: routing to another LAN

- 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



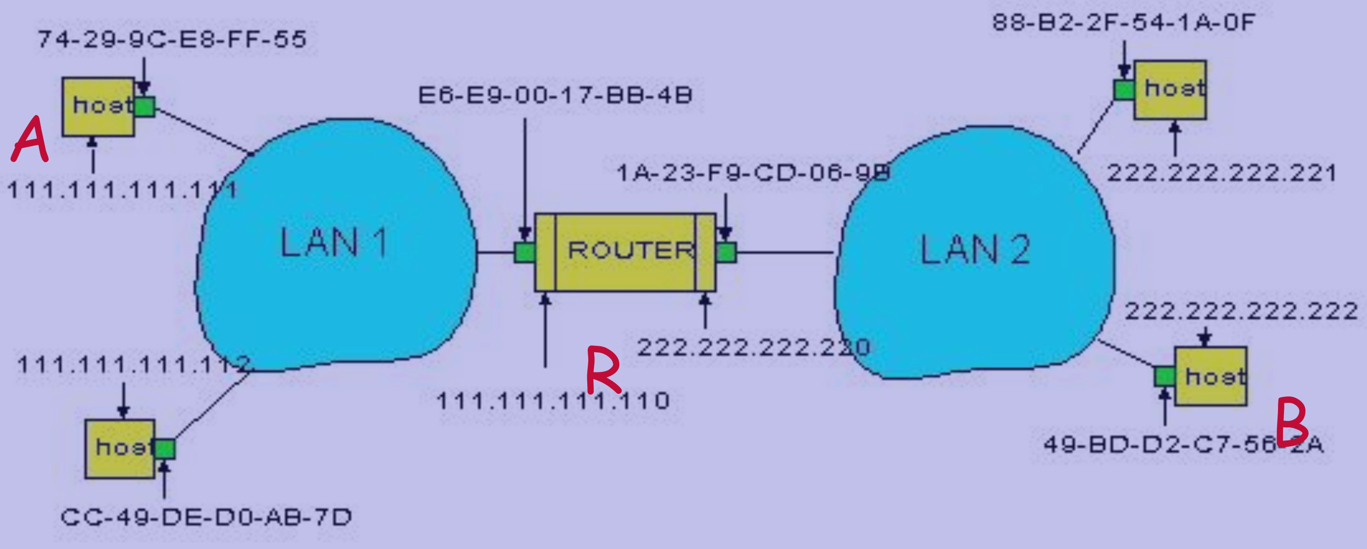
Routing to another LAN

send datagram from A to B via R
assume A know's B IP address



- Two ARP tables in router R, one for each IP network (LAN)

- A creates IP datagram with 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
- A's adapter sends frame
- R's adapter receives frame
- R removes IP datagram from Ethernet frame, sees its destined to B
- R uses ARP to get B's MAC address
- R creates frame containing A-to-B IP datagram sends to B



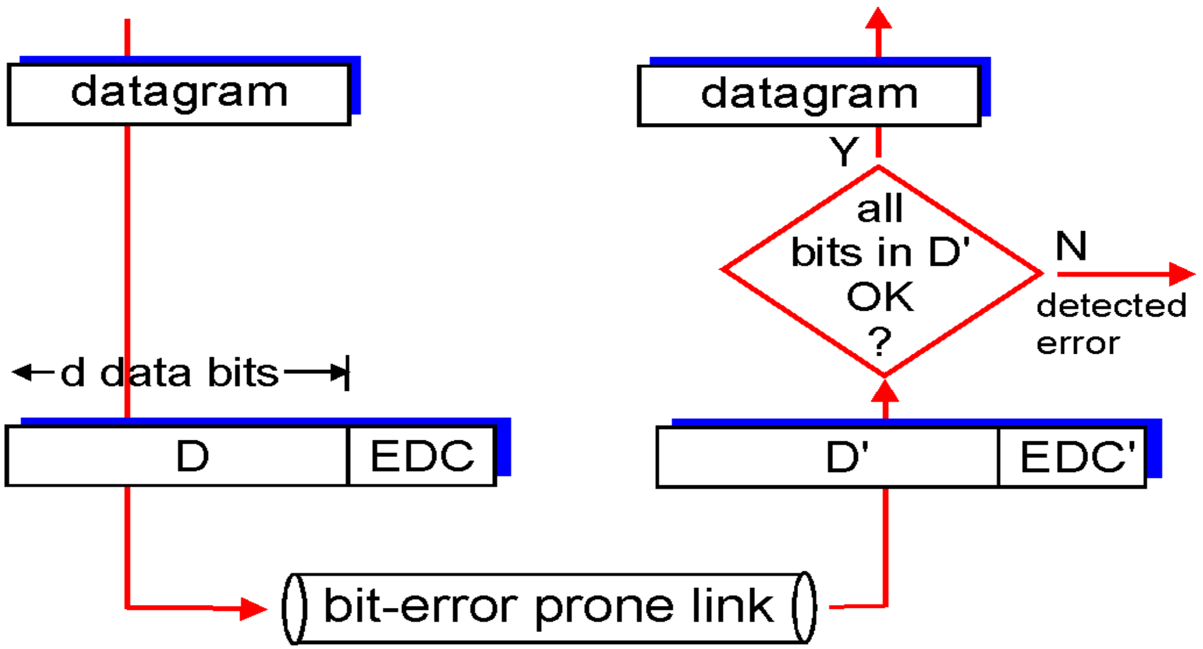
Link Layer

- Introduction and services
- Error detection and correction
- Multiple access protocols
- Link-Layer Addressing
- Ethernet
- Hubs and switches
- PPP
- MPLS

Error Detection

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

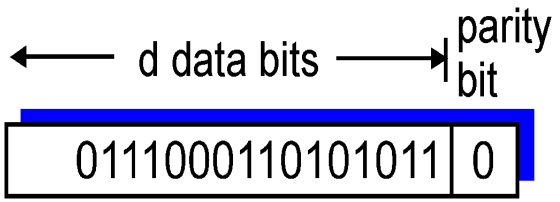
- Error detection not 100% reliable!
    - protocol may miss some errors, but rarely
    - larger EDC field yields better detection and correction



Parity Checking

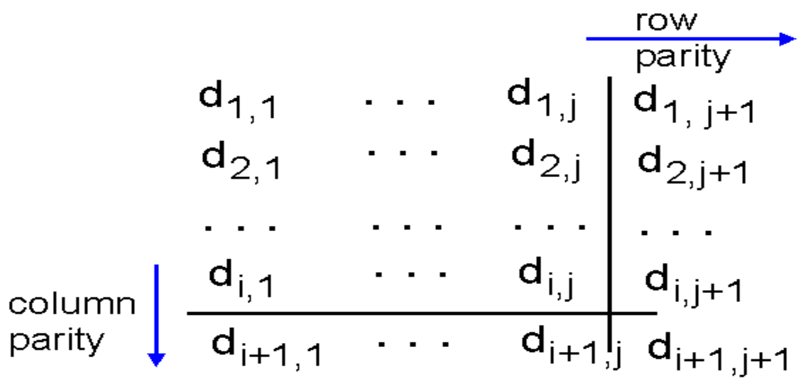
Single Bit Parity:

Detect single bit errors

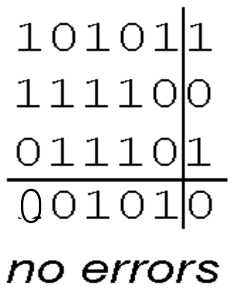


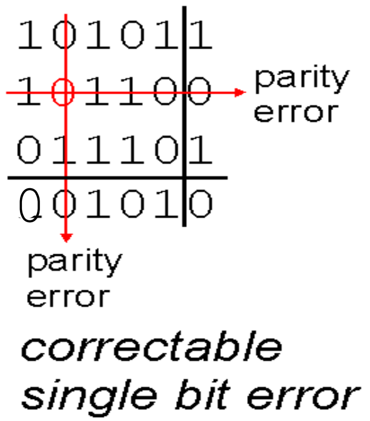
Two Dimensional Bit Parity:

Detect *and correct* single bit errors



row parity
column parity
$d_{1,1}$ . . . $d_{1,j}$ $d_{1,j+1}$
$d_{2,1}$ . . . $d_{2,j}$ $d_{2,j+1}$
. . . .
$d_{i,1}$ . . . $d_{i,j}$ $d_{i,j+1}$
$d_{i+1,1}$ . . . $d_{i+1,j}$ $d_{i+1,j+1}$





Cyclic Redundancy Check (CRC) --- Polynomial Codes

- A (n+1)-bit message can be represented as a polynomial of degree n. For example,
  - X = 10011010;
  - M(X) = $x^7 + x^4 + x^3 + x$
- So, a sender and receiver can be considered to exchange polynomials (in binary).
- Choose k+1 bit pattern (divisor), C(X), a polyn of degree k
- goal: choose k CRC bits, R, such that
  - <M,R> exactly divisible by C (modulo 2)
  - receiver knows C, divides <M,R> by C. If non-zero remainder: error detected!
  - can detect all burst errors less than k+1 bits

CRC continued …

- Goal: design P(X) such that it is exactly divisible by C(X)
- Multiply M(X) by x^k (add k zero's to the end of the message) to get T(X)
- Divide T(X) by C(X) and find the remainder R(X)
- Subtract the remainder from T(X) to get P(X). P(X) is now exactly divisible by C(X).

- Remember – all addition/subtract use modulo-2 arithmetic.

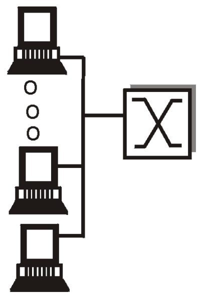
Link Layer

- Introduction and services
- Error detection and correction
- Multiple access protocols
- Link-Layer Addressing
- Ethernet
- Hubs and switches
- PPP
- MPLS

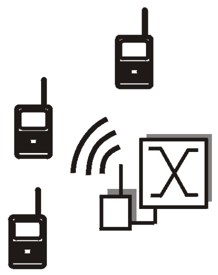
Multiple Access Links and Protocols

Two types of "links":

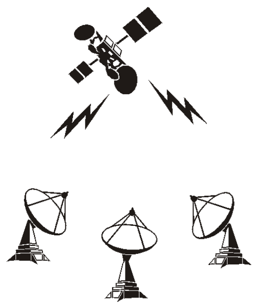
- point-to-point
  - PPP for dial-up access
  - point-to-point link between Ethernet switch and host
- broadcast (shared wire or medium)
  - traditional Ethernet
  - upstream HFC
  - 802.11 wireless LAN



shared wire
(e.g. Ethernet)



shared wireless
(e.g. Wavelan)



satellite

cocktail party

When are Multiple Access Protocols Required?

- single shared broadcast channel
- two or more simultaneous transmissions by nodes:
  - collision if node receives two or more signals at the same time (examples, LANs, Wireless-LANs)

When are Multiple Access Protocols Required?

- single shared broadcast channel
- two or more simultaneous transmissions by nodes:
  - collision if node receives two or more signals at the same time (examples, LANs, Wireless-LANs)

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

Ideal Multiple Access Protocol

Broadcast channel of rate R bps

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 and easy to implement

Taxonomy of Multiple Access Control Protocols

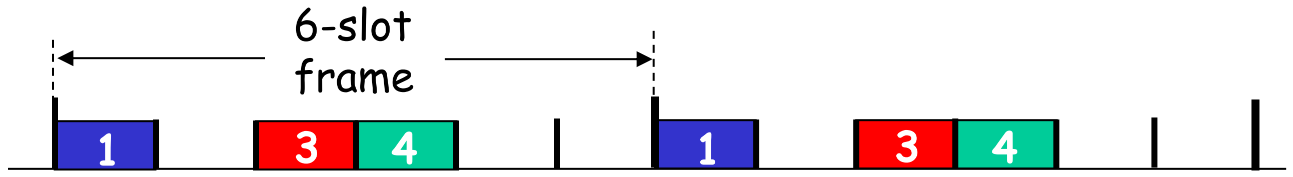
Three broad classes:

- 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
- "Taking turns"
  - nodes take turns, but nodes with more to send can take longer turns

Channel Partitioning MAC protocols: TDMA

TDMA: time division multiple access

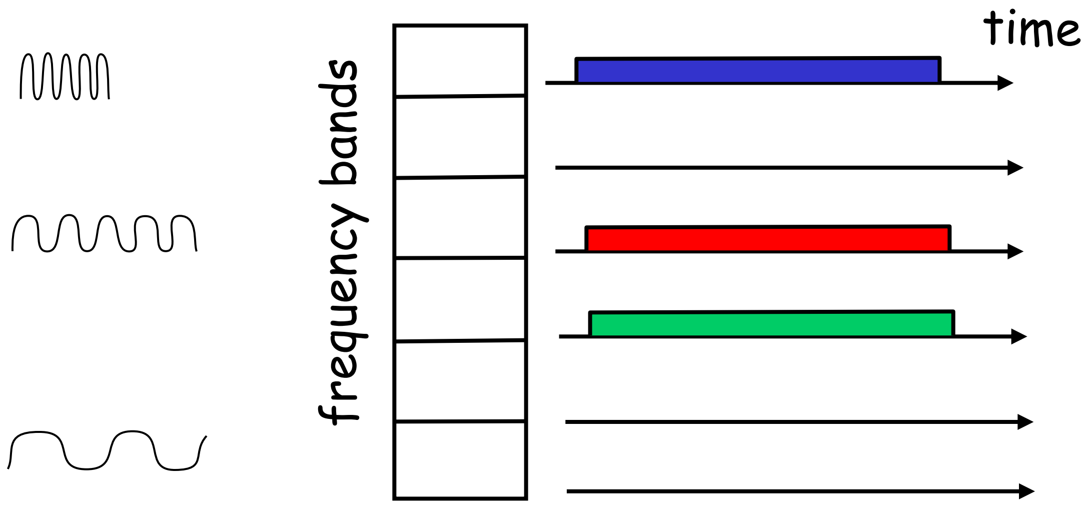
- access to channel in "rounds"
- each station gets fixed length slot (length = pkt trans time) in each round
- unused slots go idle
- Example 6-station LAN:
  - 1,3,4 have pkt, slots 2,5,6 idle



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 pkt, frequency bands 2,5,6 idle



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 leads to "collision"

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 leads to "collision"
- random access MAC protocol specifies:
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed retransmissions)

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 leads to "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:
  - pure ALOHA
  - slotted ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA

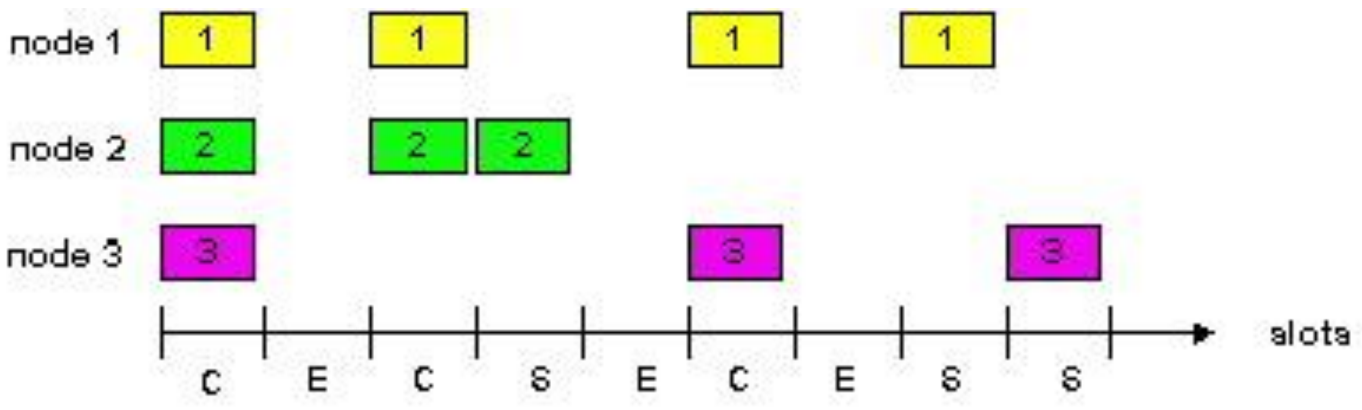
Assumptions

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

Operation

- when node obtains fresh frame, it transmits in next slot
- 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: 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 is the long-run fraction of successful slots when there are many nodes, each with many frames to send

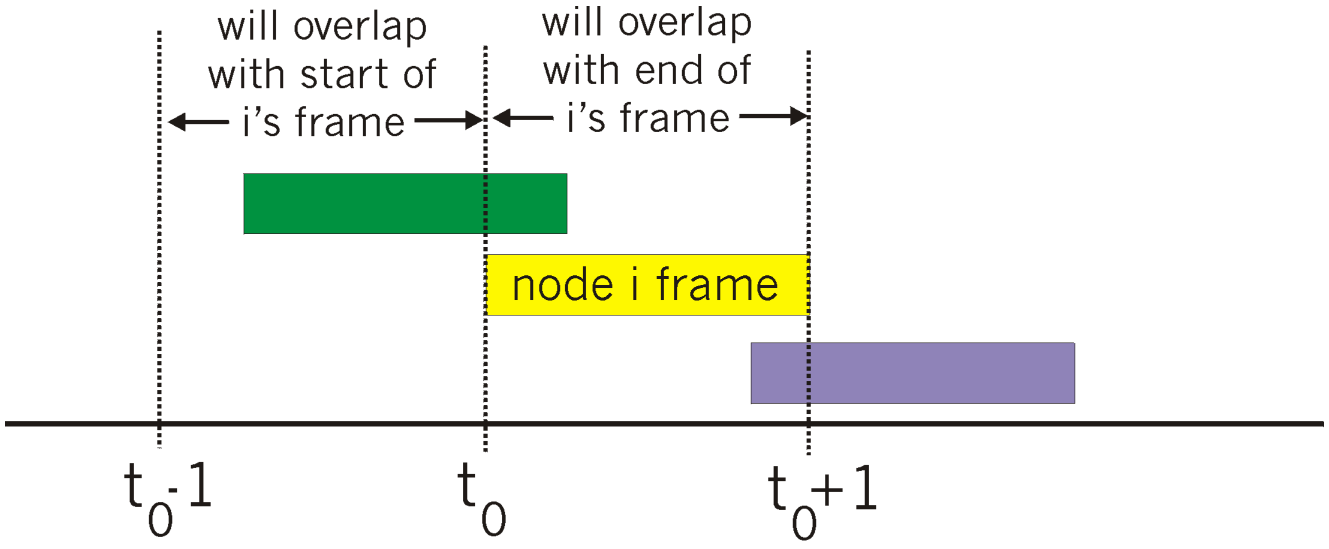
- Suppose N nodes with many frames to send, each transmits in slot with probability *p*
- prob that node 1 has success in a slot $= p(1-p)^{N-1}$
- prob that any node has a success $= Np(1-p)^{N-1}$

- For max efficiency with N nodes, 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 1/e = .37

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

Pure ALOHA

- Let nodes transmit whenever a frame is ready
- No synchronization among nodes
  - If collision, retransmit after random delay
- 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 $[p_0\text{-}1,p_0]$ ·

P(no other node transmits in $[p_0\text{-}1,p_0]$

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

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

… choosing optimum p and then letting n -> infty ...

Pretty lousy!

= 1/(2e) = .18

CSMA (Carrier Sense Multiple Access)

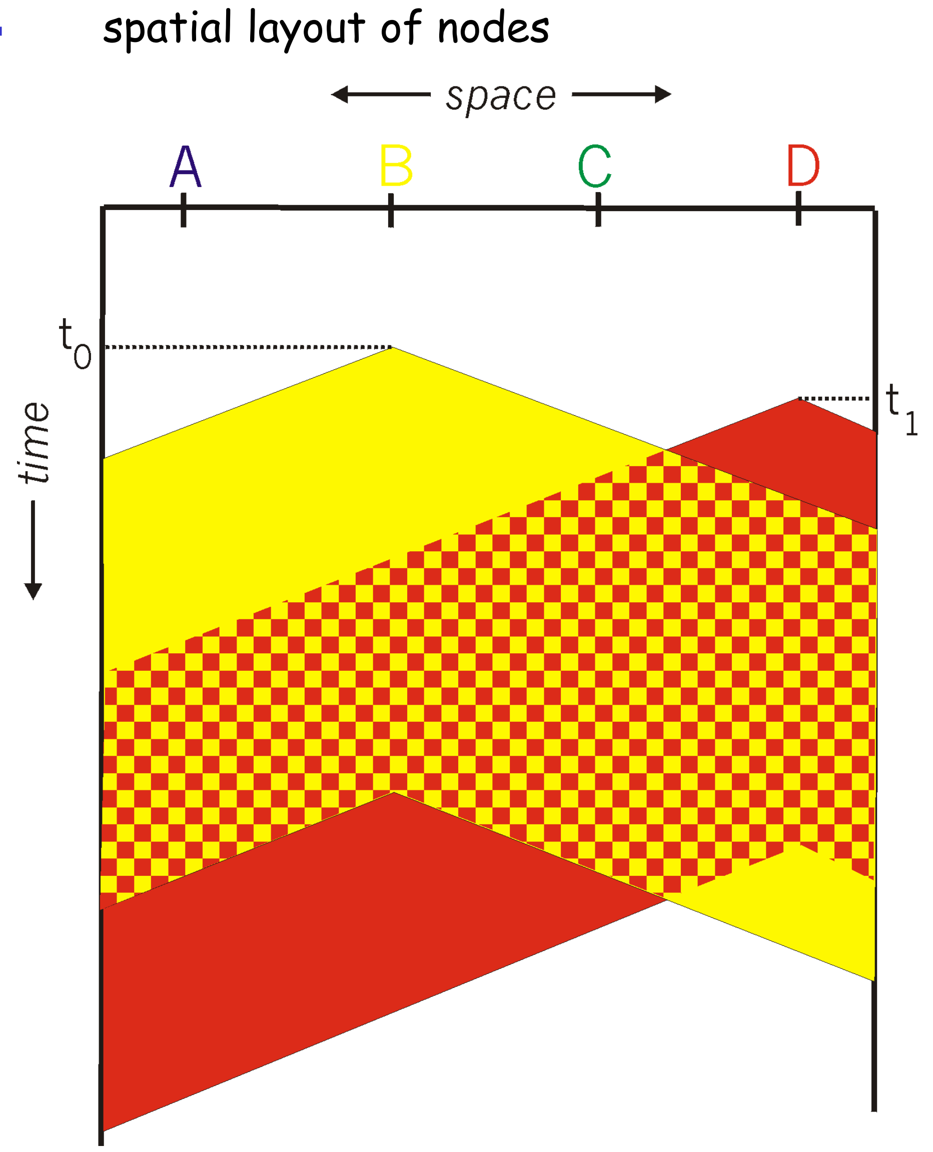
CSMA: listen before transmit:

If channel sensed idle: transmit entire frame

- If channel sensed busy, defer transmission

- human analogy: don't interrupt others!

CSMA collisions



collisions can still occur:
propagation delay means two nodes may not hear each other's transmission

collision:
entire packet transmission time wasted

note:
role of distance & propagation delay in determining collision probability

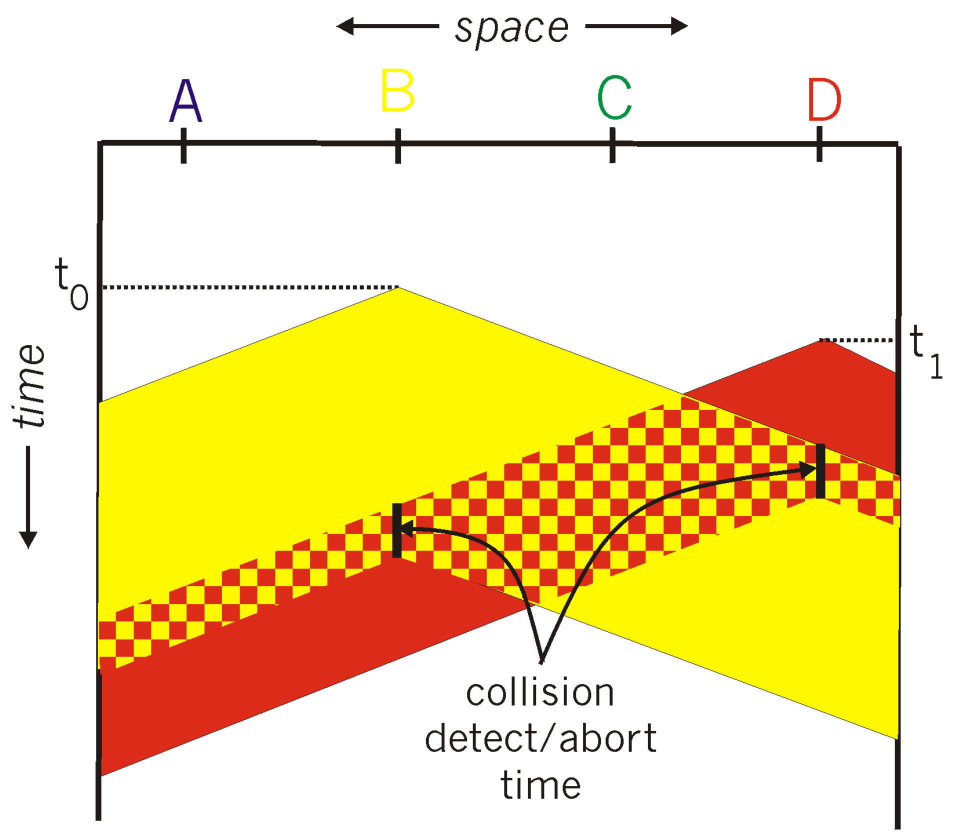
CSMA/CD (Collision Detection)

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: receiver shut off while transmitting (See CSMA/CA in Ch 6 instead)
- human analogy: the polite conversationalist

CSMA/CD collision detection



“Turn-Taking” MAC protocols

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!

Random access MAC protocols

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

“Turn-taking” protocols

look for best of both worlds!

“Taking Turns” MAC protocols

Polling:

- master node “invites” slave nodes to transmit in turn

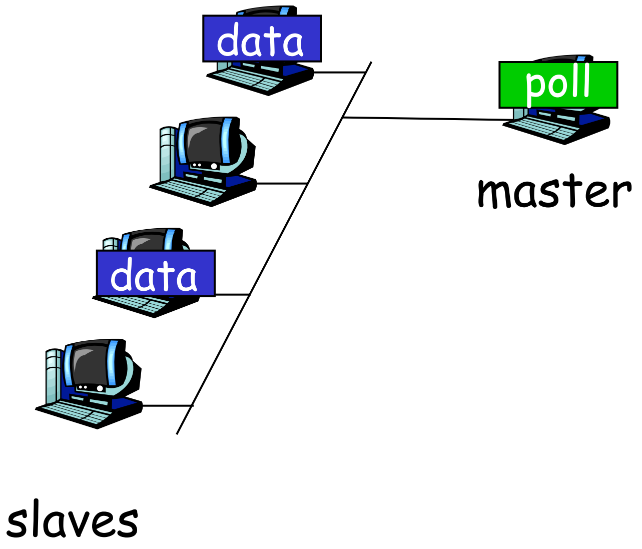
Token passing:

- control **token** passed from one node to next sequentially.

"Taking Turns" MAC protocols

Polling:

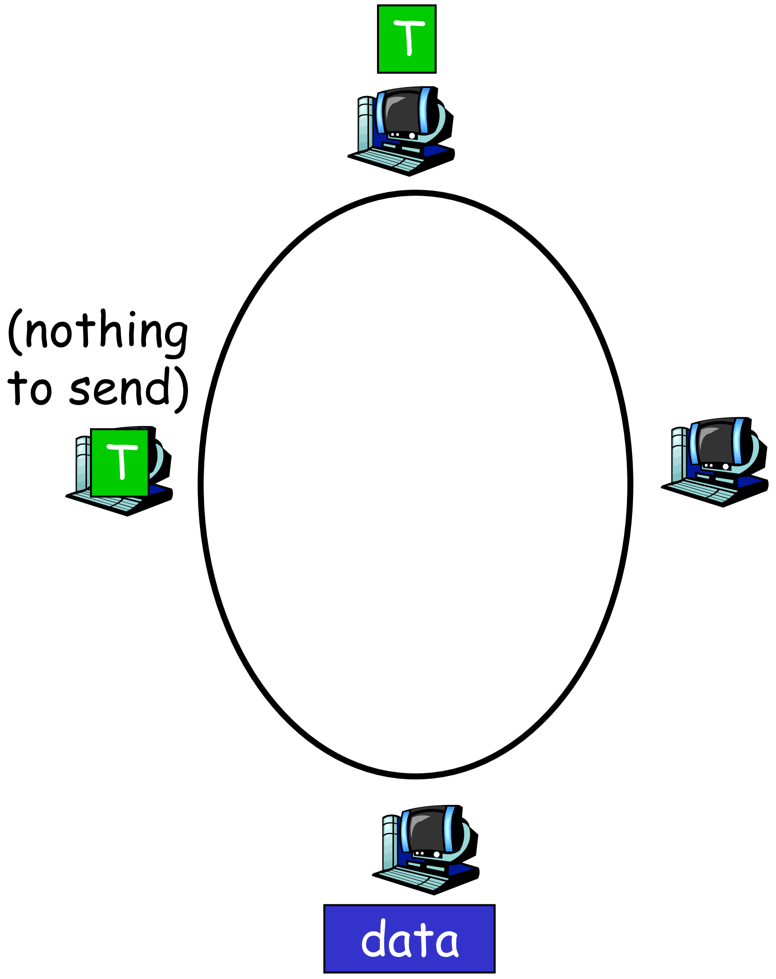
- master node "invites" slave nodes to transmit in turn
- typically used with "dumb" slave devices
- concerns:
  - polling overhead
  - latency
  - single point of failure (master)



"Taking Turns" MAC protocols

Token passing:

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



Summary of MAC protocols

- What do you do with a shared media?
  - Channel Partitioning, by time, frequency or code
    - Time Division, Frequency Division, Code Division
  - Random partitioning (dynamic),
    - ALOHA, Slotted ALOHA, CSMA, CSMA/CD
    - carrier sensing: easy in some technologies (wire), hard in others (wireless)
    - CSMA/CD used in Ethernet (more below)
    - CSMA/CA used in 802.11 wireless LANs (next lecture)
  - Turn-Taking
    - polling, token passing, token ring, FDDI

Next Stop LAN Technologies

Data link layer so far:

- services, error detection/correction, multiple access

Next: LAN technologies

- Ethernet
- hubs, switches
- PPP

Link Layer

- Introduction and services
- Error detection and correction
- Multiple access protocols
- Link-Layer Addressing
- Ethernet
- Hubs and switches
- PPP
- MPLS

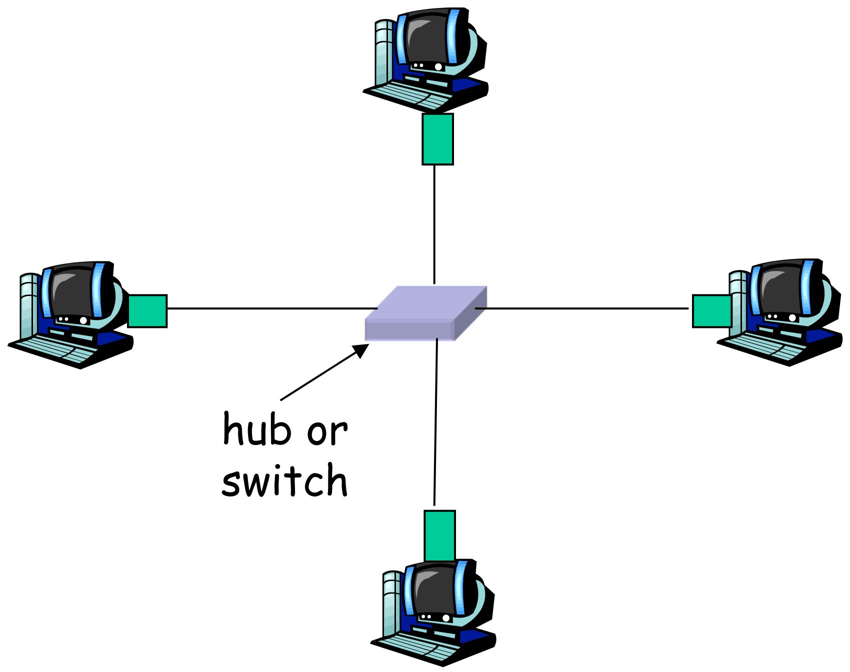
Ethernet

"dominant" wired LAN technology:

- cheap $20 for 100 Mbps!
- first widely used LAN technology
- Simpler, cheaper than token LANs and ATM
- Kept up with speed race: 10 Mbps – 10 Gbps

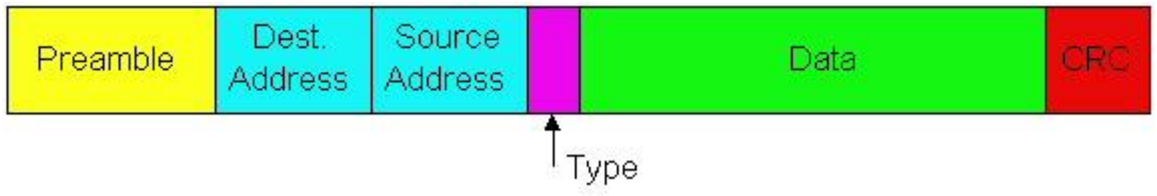
Star topology

- Bus topology popular through mid 1990s
- Now star topology prevails
- Connection choices: hub or switch (more later)



Ethernet Frame Structure (1/2)

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

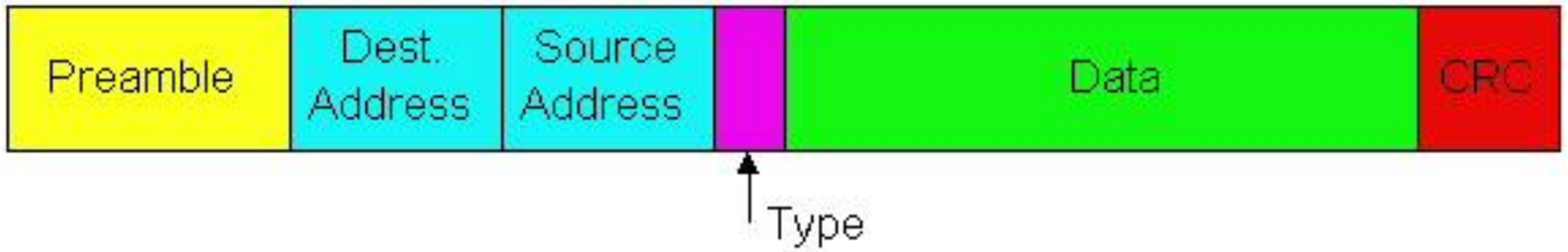


Preamble:

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

Ethernet Frame Structure (2/2)

- Addresses: 6 bytes
  - if adapter receives frame with matching destination address, or with broadcast address (eg ARP packet), it passes data in frame to net-layer protocol
  - otherwise, adapter discards frame
- Type: indicates the higher layer protocol (mostly IP but others may be supported such as Novell IPX and AppleTalk)
- CRC: checked at receiver, if error is detected, the frame is simply dropped



Unreliable, connectionless service

- Connectionless: No handshaking between sending and receiving adapter.
- Unreliable: receiving adapter doesn't send acks or nacks to sending adapter
  - stream of datagrams passed to network layer can have gaps
  - gaps will be filled if app is using TCP
  - otherwise, app will see the gaps

Ethernet uses CSMA/CD

- No slots
- Adapter doesn't transmit if it senses that some other adapter is transmitting, that is, carrier sense
- Transmitting adapter aborts when it senses that another adapter is transmitting, that is, collision detection
- Before attempting a retransmission, adapter waits a random time, that is, random access

Ethernet CSMA/CD algorithm

1. Adaptor receives datagram from net layer & creates frame
2. If adapter senses channel idle, it starts to transmit frame. If it senses channel busy, waits until channel idle and then transmits
3. If adapter transmits entire frame without detecting another transmission, the adapter is done with frame !
4. If adapter detects another transmission while transmitting, aborts and sends jam signal
5. After aborting, adapter enters **exponential backoff**: after the mth collision, adapter chooses a K at random from $\{0,1,2,\ldots,2^m-1\}$. Adapter waits K 512-bit times and returns to Step 2

Ethernet's CSMA/CD (more)

Jam Signal: make sure all other transmitters are aware of collision; 48 bits

Bit time: .1 microsec for 10 Mbps Ethernet ;
for K=1023, wait time is about 50 msec

See/interact with Java applet on book Web site: highly recommended !

Exponential Backoff:

- *Goal*: adapt retransmission attempts to estimated current load
  - heavy load: random wait will be longer
- first collision: choose K from {0,1}; delay is K x 512 bit transmission times
- after second collision: choose K from {0,1,2,3}…
- after ten collisions, choose K from {0,1,2,3,4,…,1023}

CSMA/CD efficiency

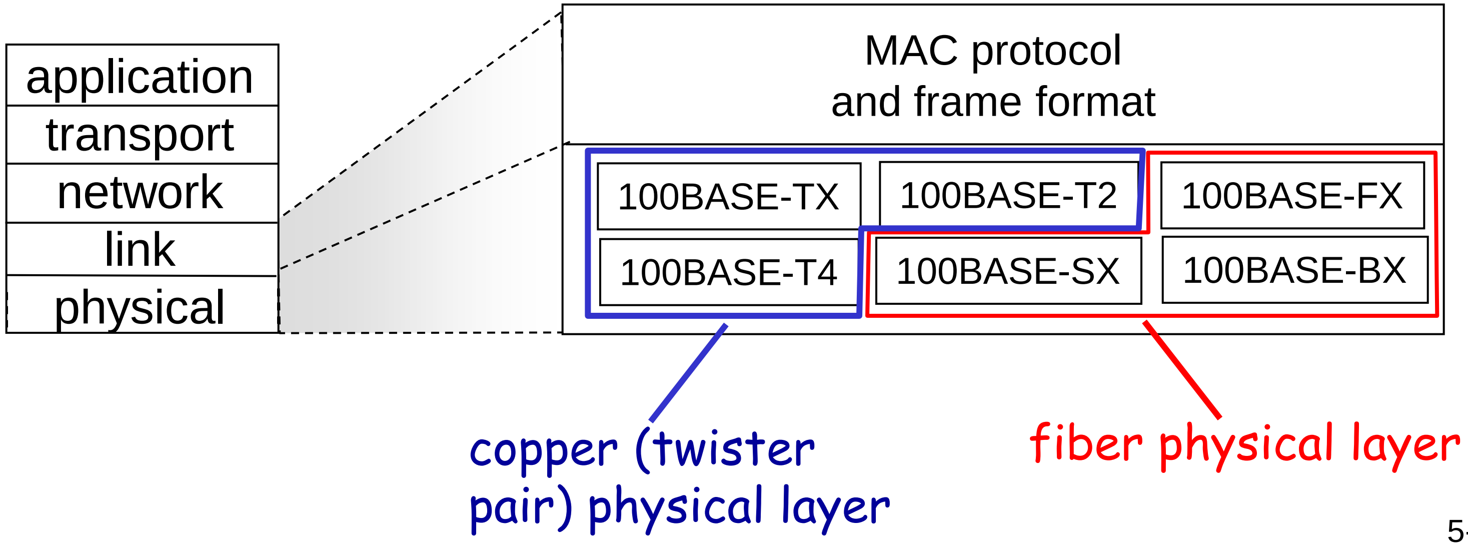
- t_{prop} = max prop between 2 nodes in LAN
- t_{trans} = time to transmit max-size frame

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

- Efficiency goes to 1 as t_{prop} goes to 0
- Goes to 1 as t_{trans} goes to infinity
- Much better than ALOHA, but still decentralized, simple, and cheap

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



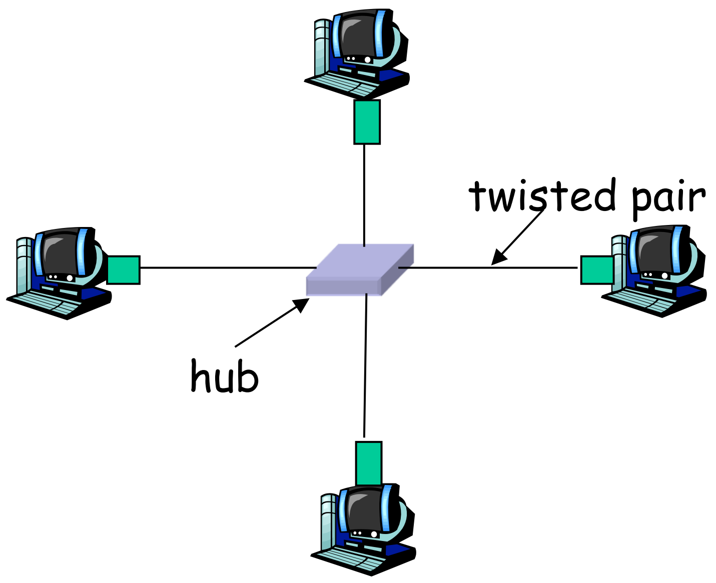
Link Layer

- Introduction and services
- Error detection and correction
- Multiple access protocols
- Link-Layer Addressing
- Ethernet
- Interconnections: Hubs and switches
- PPP
- MPLS

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

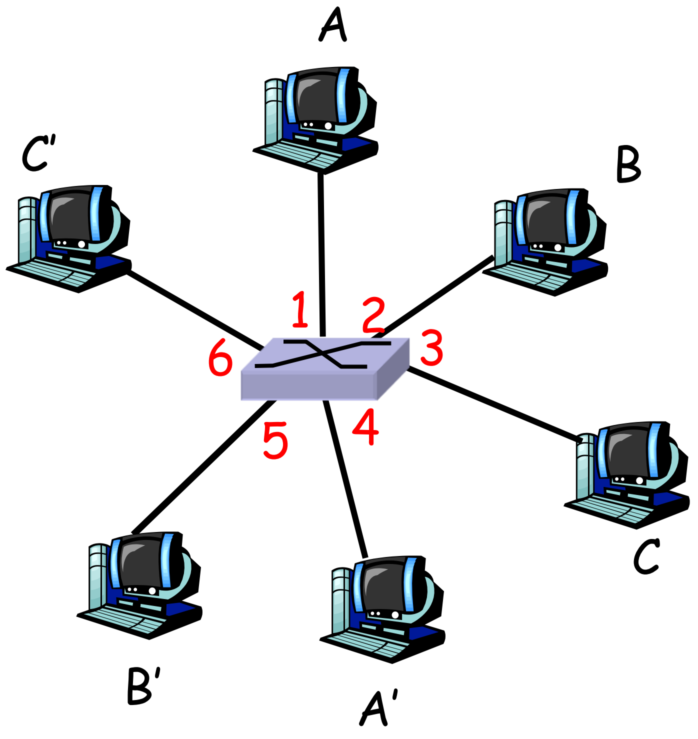


Switch

- link-layer device: smarter than hubs, take *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

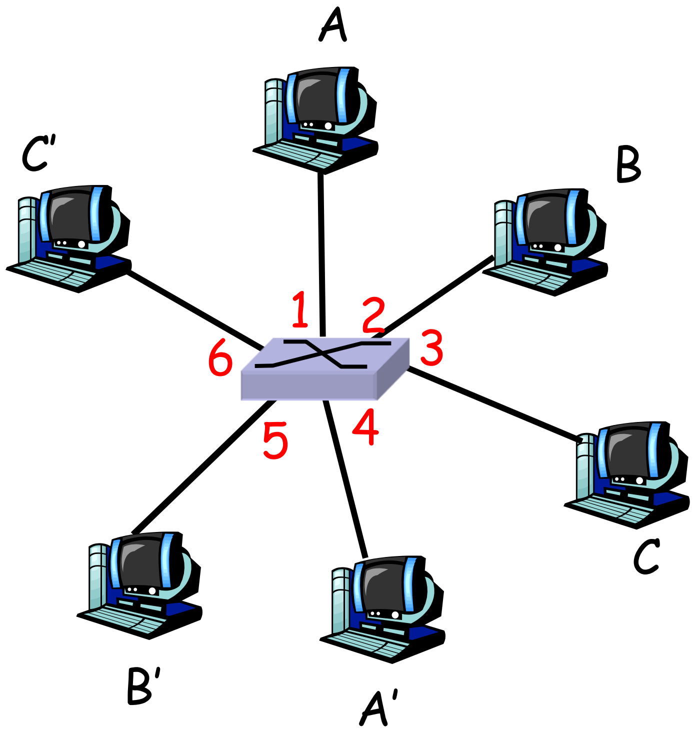
- hosts have dedicated, direct connection to switch
- switches buffer packets
- Ethernet protocol used on *each* incoming link, but no collisions; full duplex
  - each link is its own collision domain
- *switching:* A-to-A' and B-to-B' simultaneously, without collisions
  - not possible with dumb hub



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

Switch Table

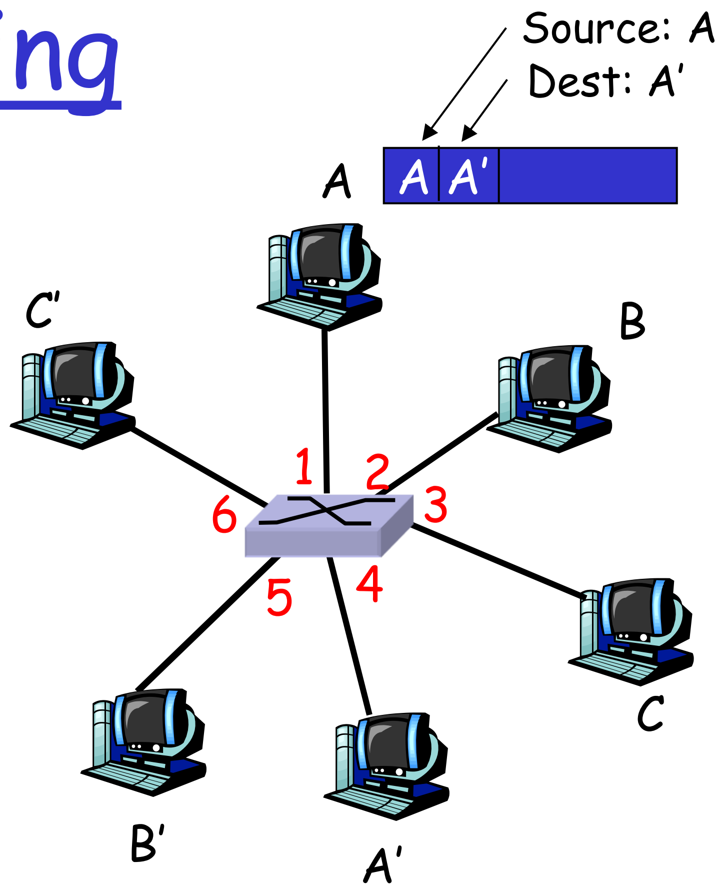
- *Q:* how does switch know that A' reachable via interface 4, B' reachable via interface 5?
- *A:* each switch has a switch table, each entry:
  - (MAC address of host, interface to reach host, time stamp)
- looks like a routing table!
- *Q:* how are entries created, maintained in switch table?



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

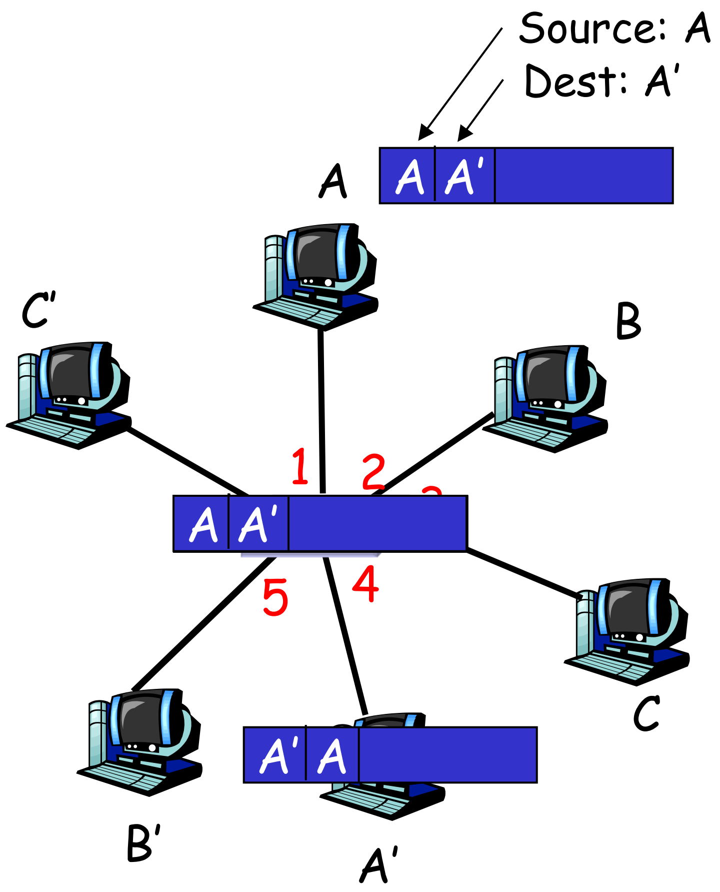
When frame received:

```
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

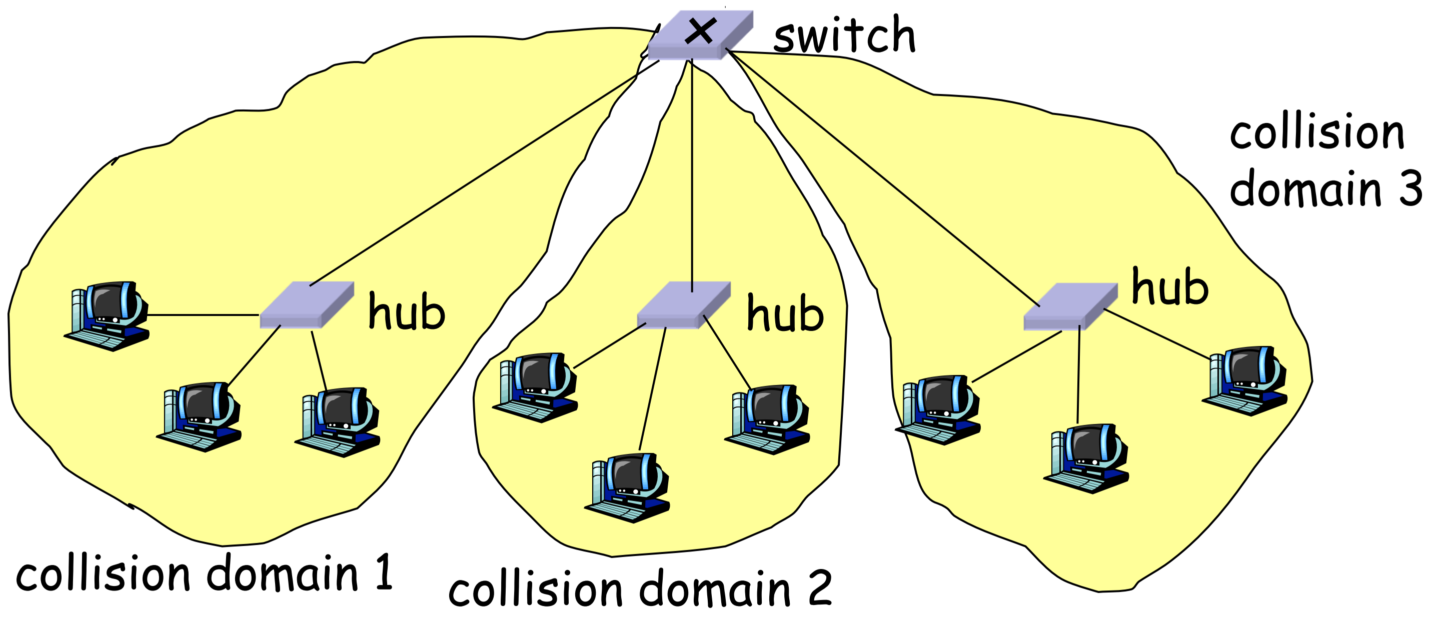


MAC addr	interface	TTL
A	1	60
A'	4	60

Switch table (initially empty)

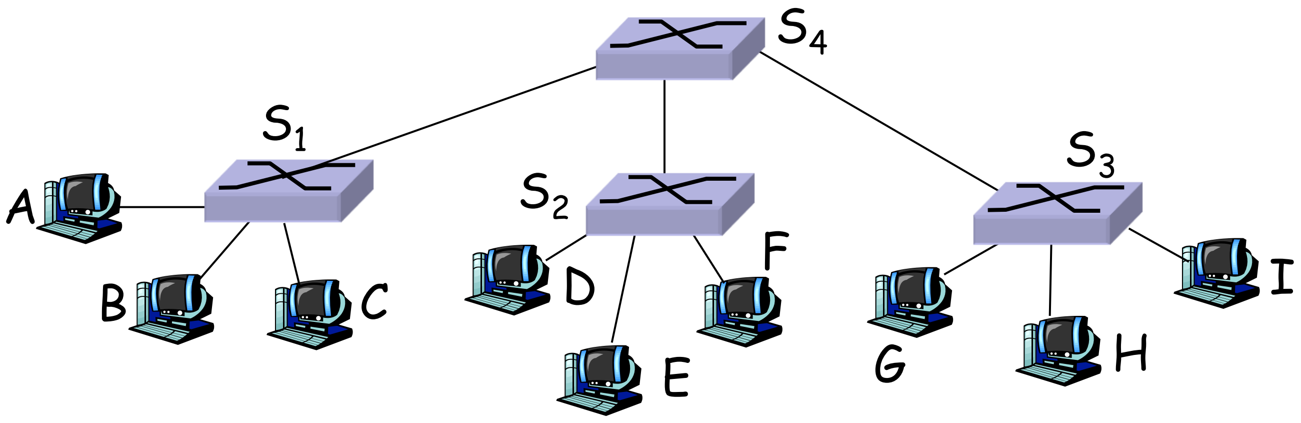
Switch: traffic isolation

- switch installation breaks subnet into LAN segments
- switch filters packets:
  - same-LAN-segment frames not usually forwarded onto other LAN segments
  - segments become separate collision domains



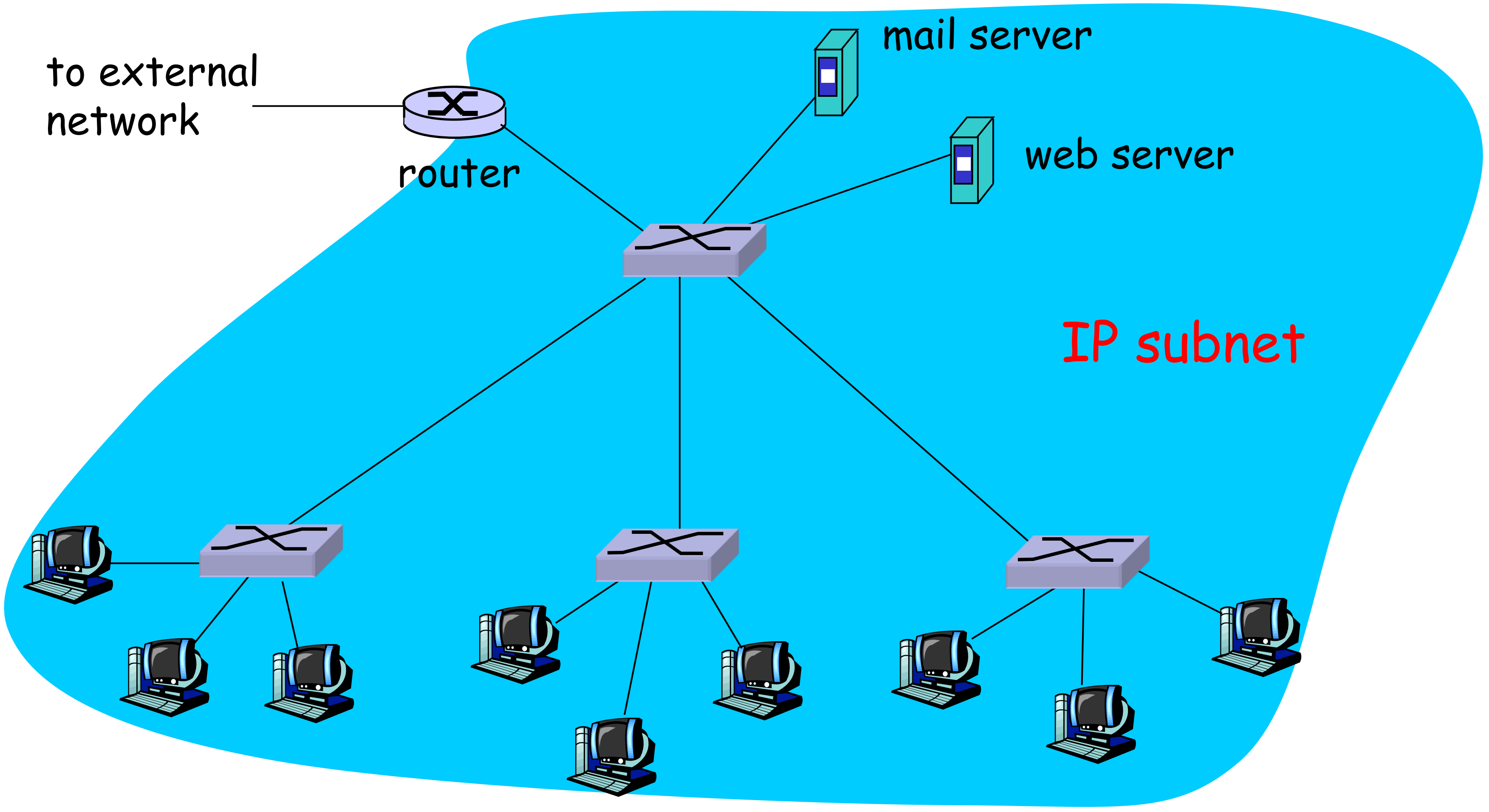
Interconnecting switches

- switches can be connected together



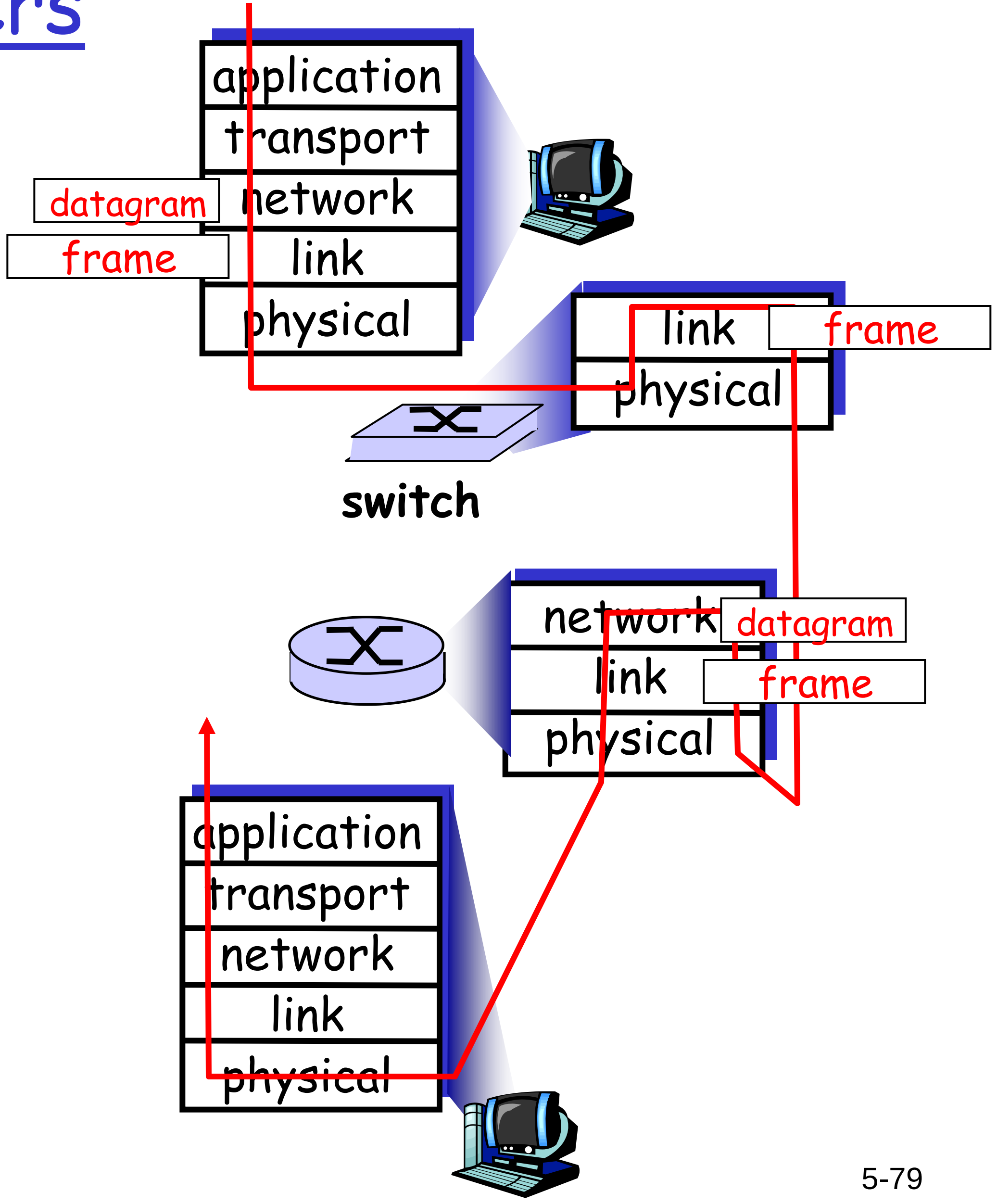
- Q: sending from A to G - how does S_1 know to forward frame destined to F via S_4 and S_3?
- A: self learning! (works exactly the same as in single-switch case!)

Institutional network

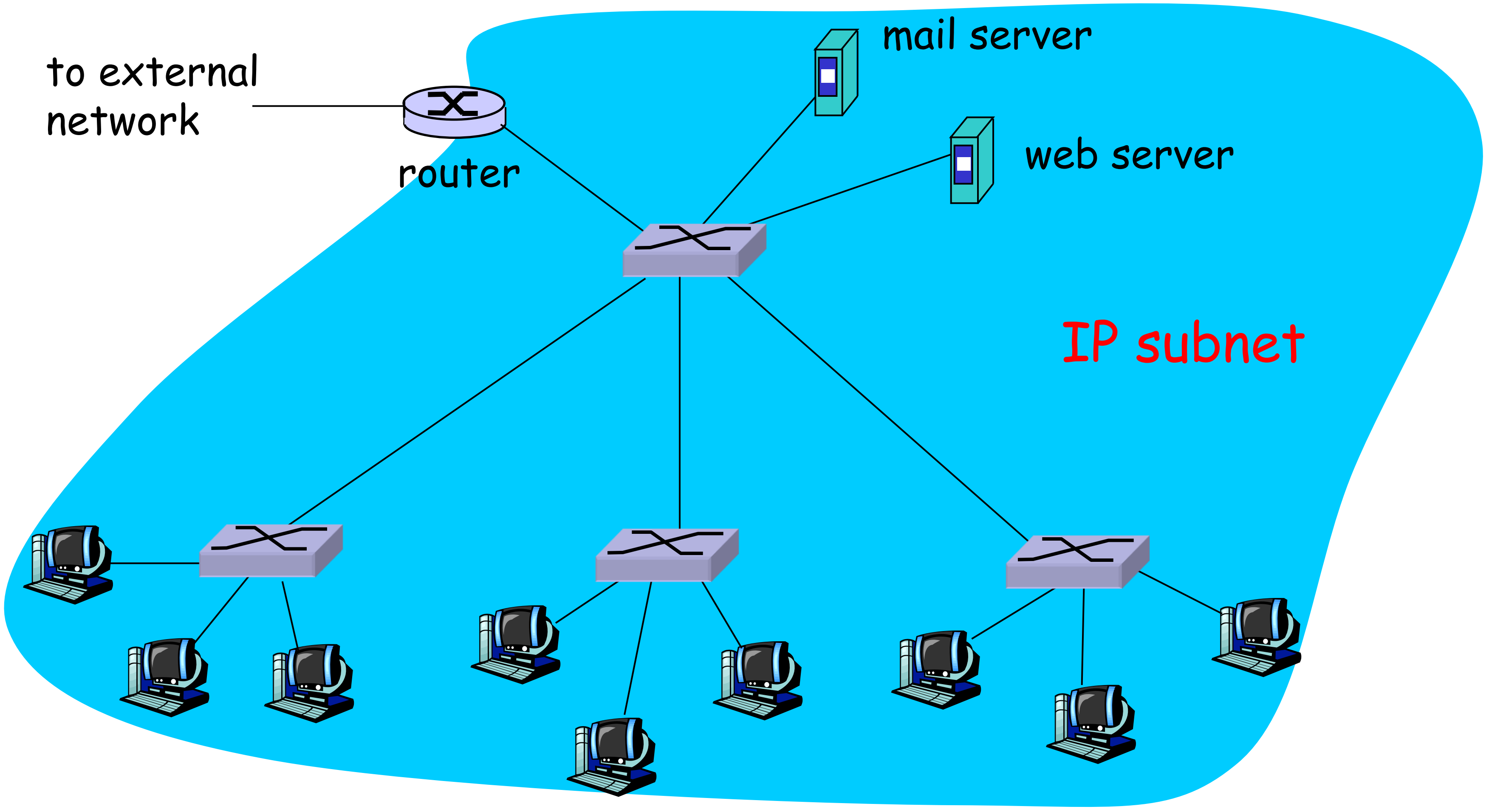


Switches vs. Routers

- both store-and-forward devices
  - Routers: network-layer devices (examine network-layer headers)
  - Switches: link-layer devices (examine link-layer headers)
- Routers maintain routing tables, implement routing algorithms
- Switches maintain switch tables, implement filtering, learning algorithms

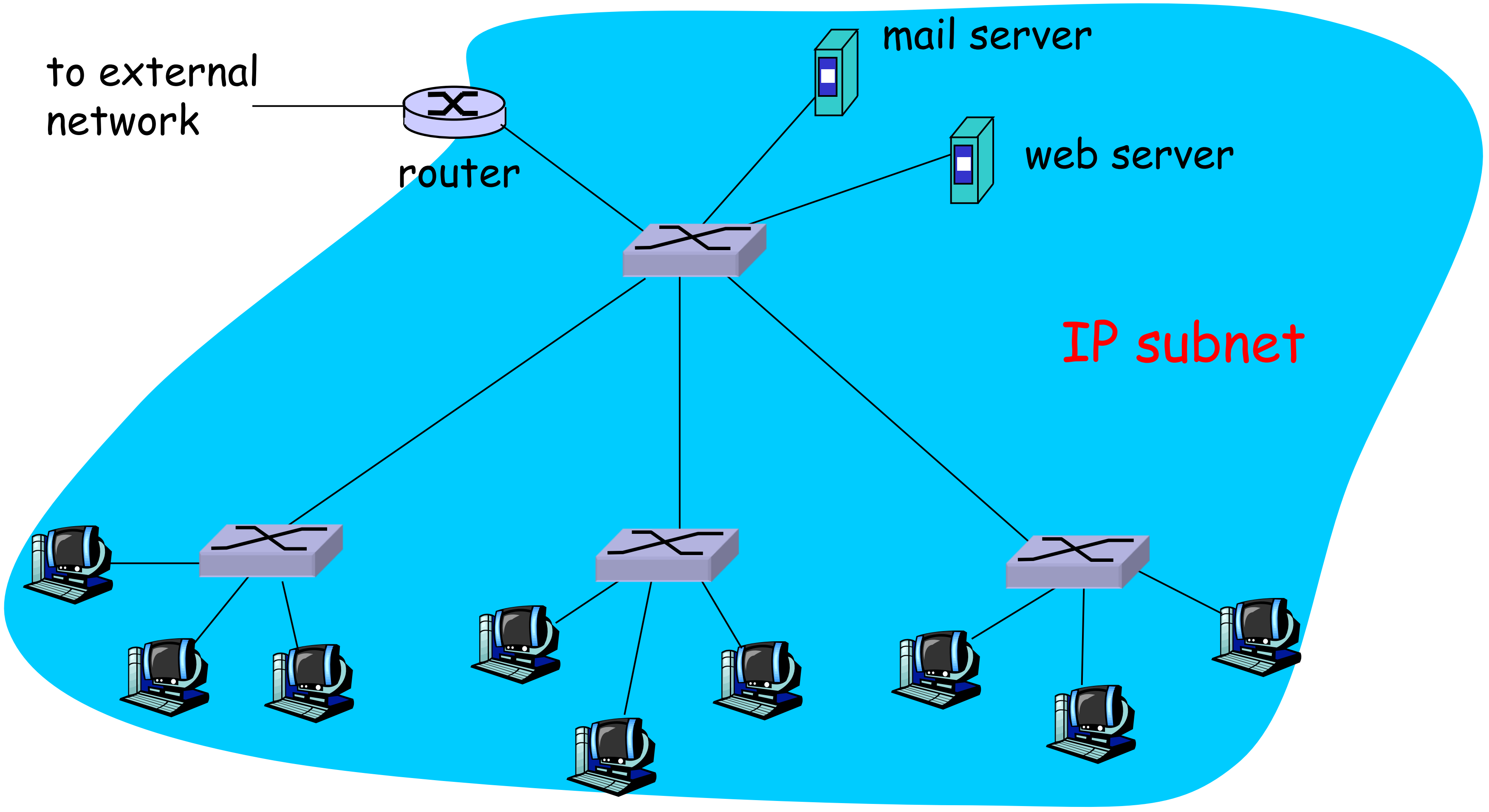


VLAN motivation



VLAN motivation

- Traffic isolation
- Inefficient use of switches
- Managing users

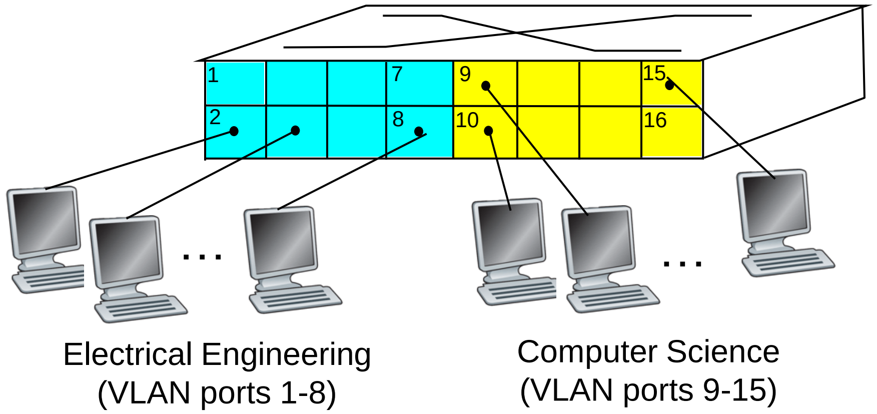


VLANs

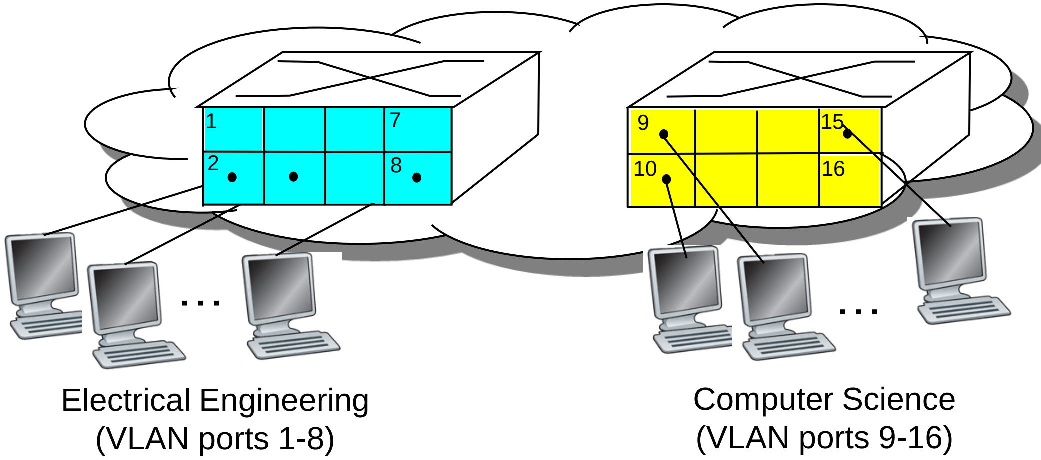
Virtual Local Area Network

Switch(es) supporting VLAN capabilities can be configured to define multiple **virtual** LANS over single physical LAN infrastructure.

Port-based VLAN: switch ports grouped (by switch management software) so that *single* physical switch ……

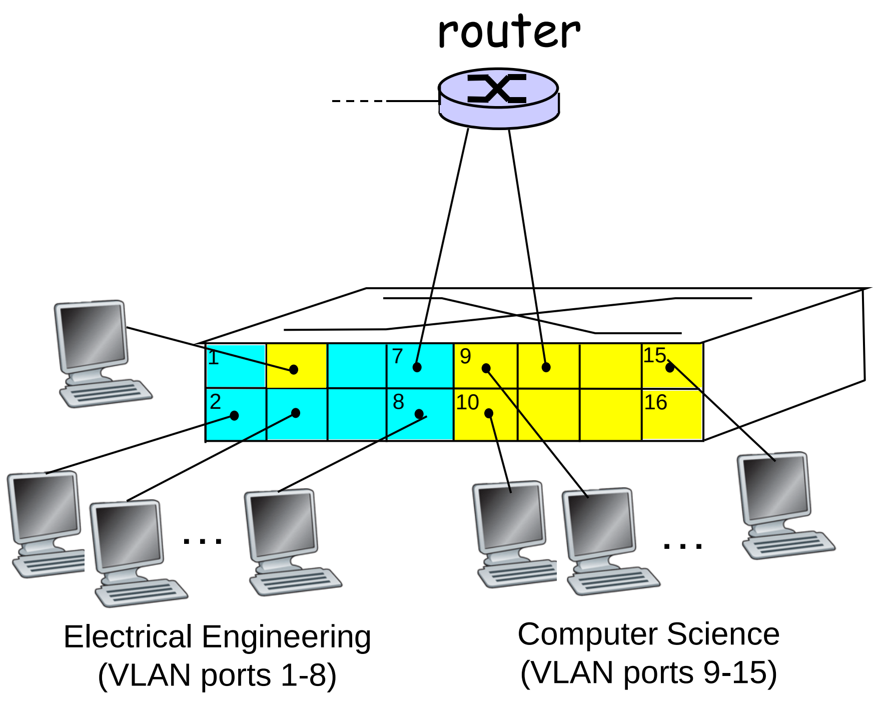


… operates as multiple virtual switches

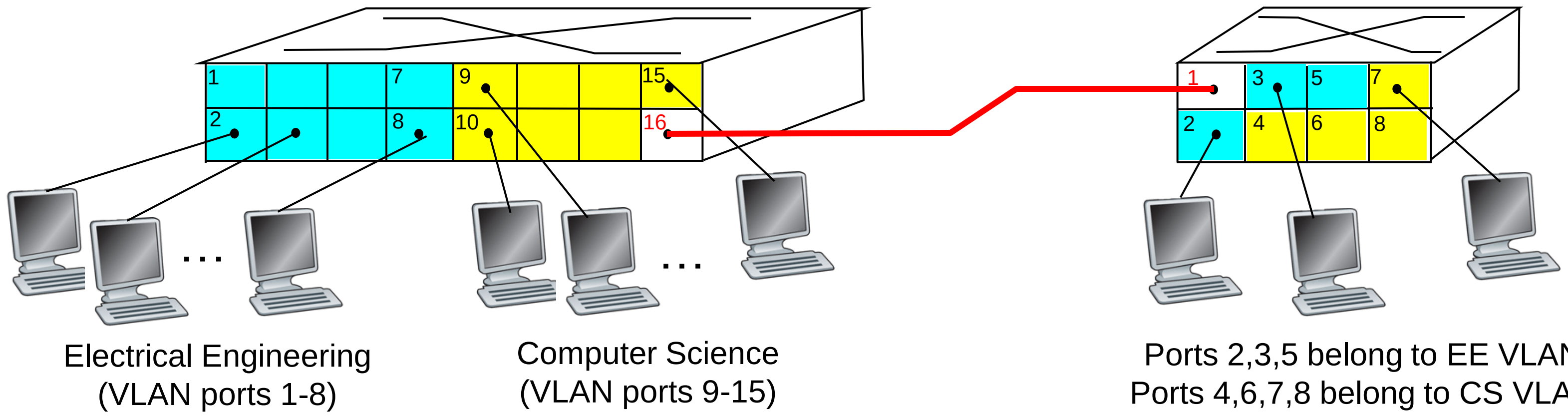


Port-based VLAN

- *traffic isolation:* frames to/from ports 1-8 can *only* reach ports 1-8
  - can also define VLAN based on MAC addresses of endpoints, rather than switch port
- dynamic membership: ports can be dynamically assigned among VLANs
- forwarding between VLANS: done via routing (just as with separate switches)
  - in practice vendors sell combined switches plus routers



VLANS spanning multiple switches



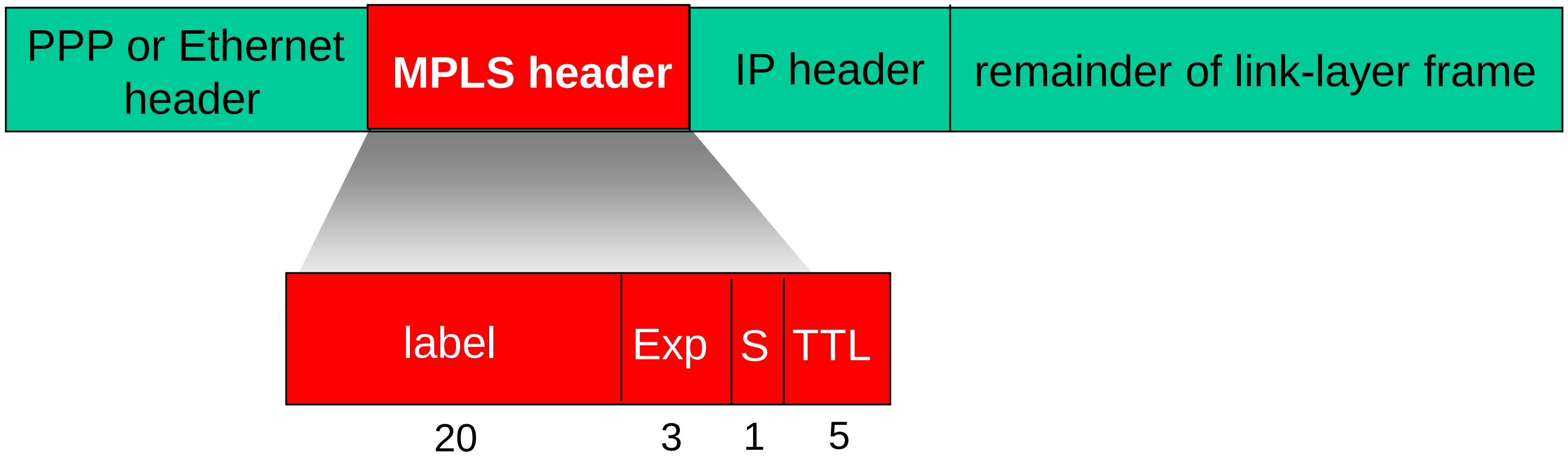
1
2
7
8
9
10
15
16
1
2
3
4
5
6
7
8
...
...
Electrical Engineering (VLAN ports 1-8)
Computer Science (VLAN ports 9-15)
Ports 2,3,5 belong to EE VLAN
Ports 4,6,7,8 belong to CS VLAN

- *trunk port:* carries frames between VLANS defined over multiple physical switches
  - frames forwarded within VLAN between switches can't be vanilla 802.1 frames (must carry VLAN ID info)
  - 802.1q protocol adds/removed additional header fields for frames forwarded between trunk ports

More slides …

Multi-Protocol Label Switching (MPLS)

- initial goal: speed up IP forwarding by using fixed length label (instead of IP address) to do forwarding
  - borrowing ideas from Virtual Circuit (VC) approach
  - but IP datagram still keeps IP address!



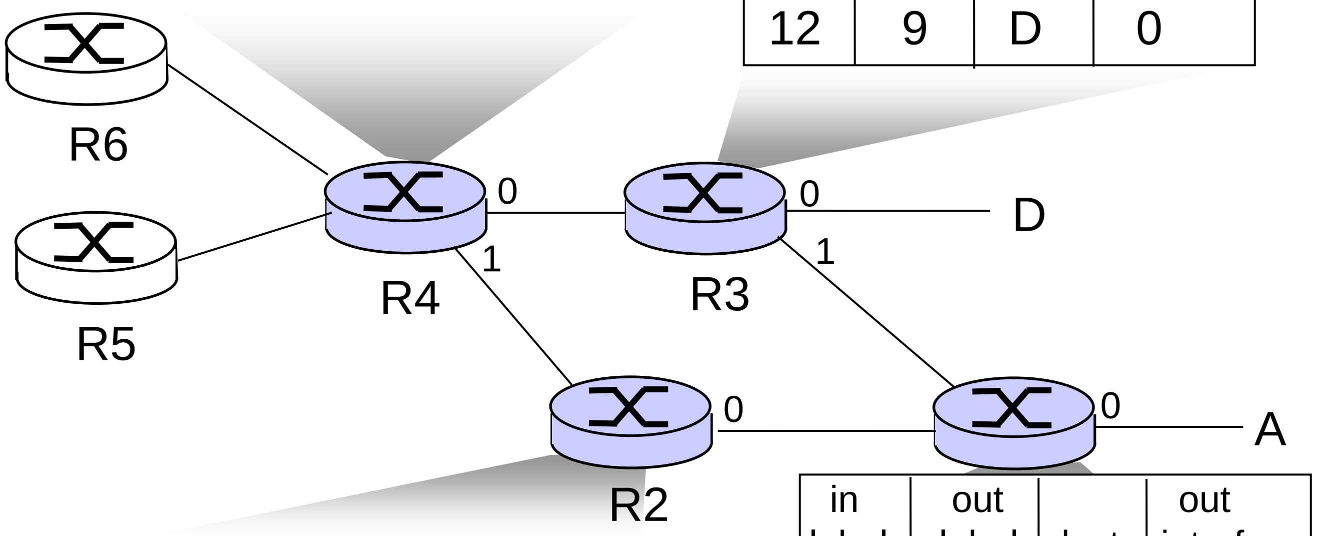
MPLS capable routers

- a.k.a. label-switched router
- forwards packets to outgoing interface based only on label value (don't inspect IP address)
  - MPLS forwarding table distinct from IP forwarding tables
- signaling protocol needed to set up forwarding
  - forwarding possible along paths that IP alone would not allow (e.g., source-specific routing) !!
  - use MPLS for traffic engineering
- must co-exist with IP-only routers

MPLS forwarding tables

in label	out label	dest	out interface
	10	A	0
	12	D	0
	8	A	1

in label	out label	dest	out interface
10	6	A	1
12	9	D	0

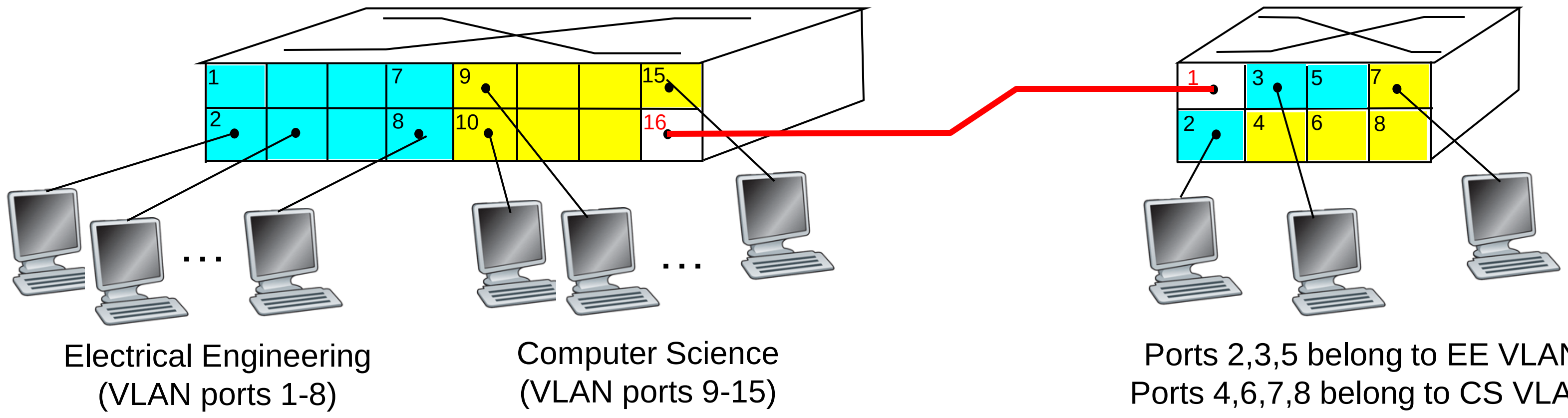


in label	out label	dest	out interface
8	6	A	0

in label	out label	dest	out interface
6	-	A	0

More slides …

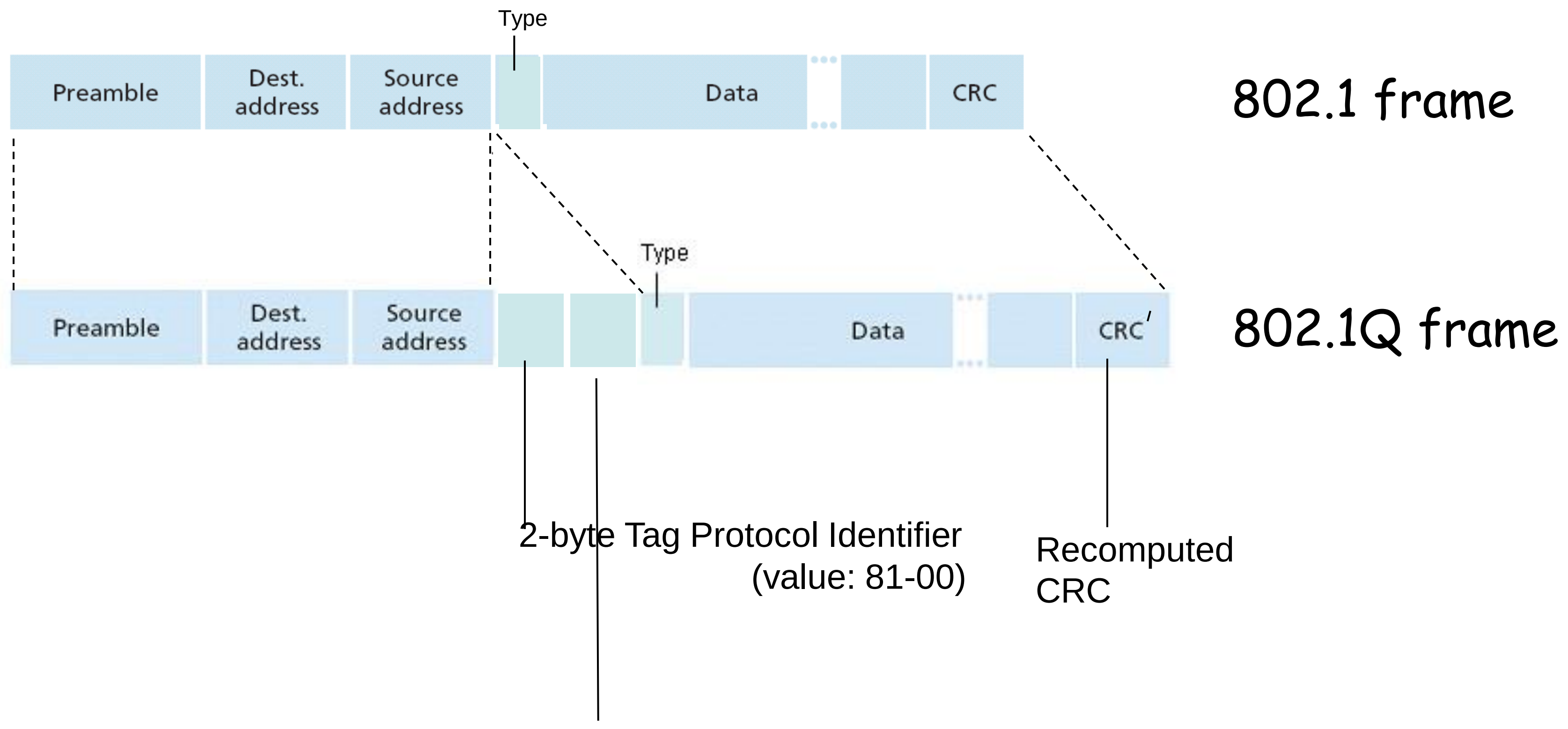
VLANS spanning multiple switches



1
2
7
8
9
10
15
16
…
…
1
2
3
4
5
6
7
8
Electrical Engineering (VLAN ports 1-8)
Computer Science (VLAN ports 9-15)
Ports 2,3,5 belong to EE VLAN
Ports 4,6,7,8 belong to CS VLAN

- *trunk port:* carries frames between VLANS defined over multiple physical switches
  - frames forwarded within VLAN between switches can't be vanilla 802.1 frames (must carry VLAN ID info)
  - 802.1q protocol adds/removed additional header fields for frames forwarded between trunk ports

802.1Q VLAN frame format



2-byte Tag Protocol Identifier (value: 81-00)

Recomputed CRC

Tag Control Information (12 bit VLAN ID field, 3 bit priority field like IP TOS)

Link Layer

- Introduction and services
- Error detection and correction
- Multiple access protocols
- Link-Layer Addressing
- Ethernet
- Hubs and switches
- PPP
- MPLS

Point to Point Data Link Control

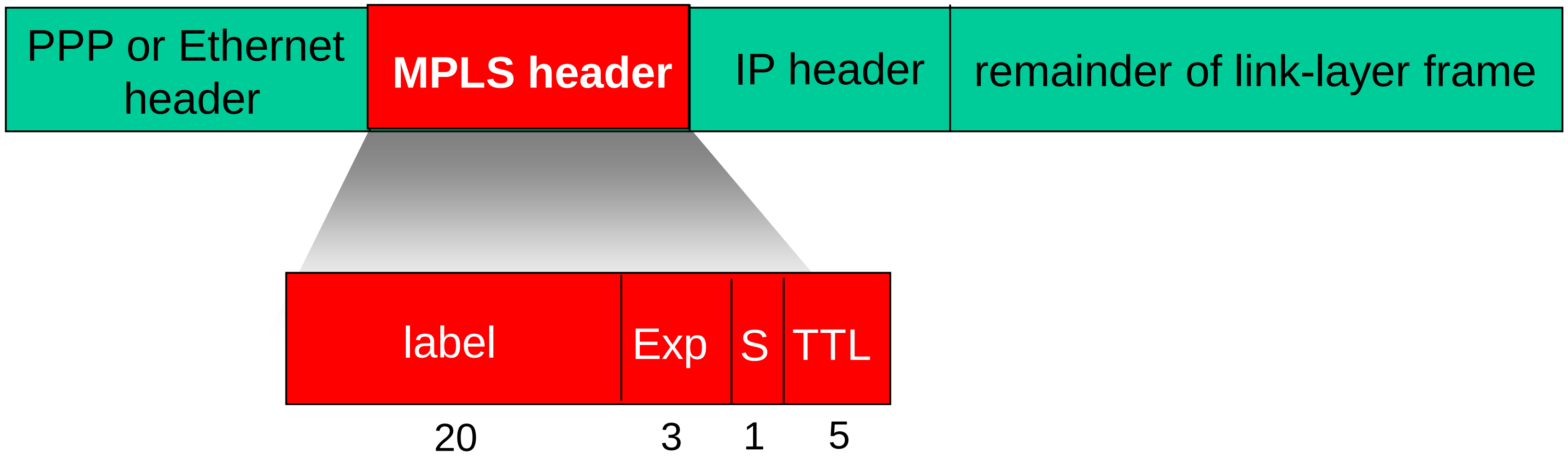
- one sender, one receiver, one link: easier than broadcast link:
  - no Media Access Control
  - no need for explicit MAC addressing
  - e.g., dialup link, ISDN line
- popular point-to-point DLC protocols:
  - PPP (point-to-point protocol)
  - HDLC: High level data link control (Data link used to be considered "high layer" in protocol stack!

Link Layer

- Introduction and services
- Error detection and correction
- Multiple access protocols
- Link-Layer Addressing
- Ethernet
- Hubs and switches
- PPP
- MPLS

Multi-Protocol Label Switching (MPLS)

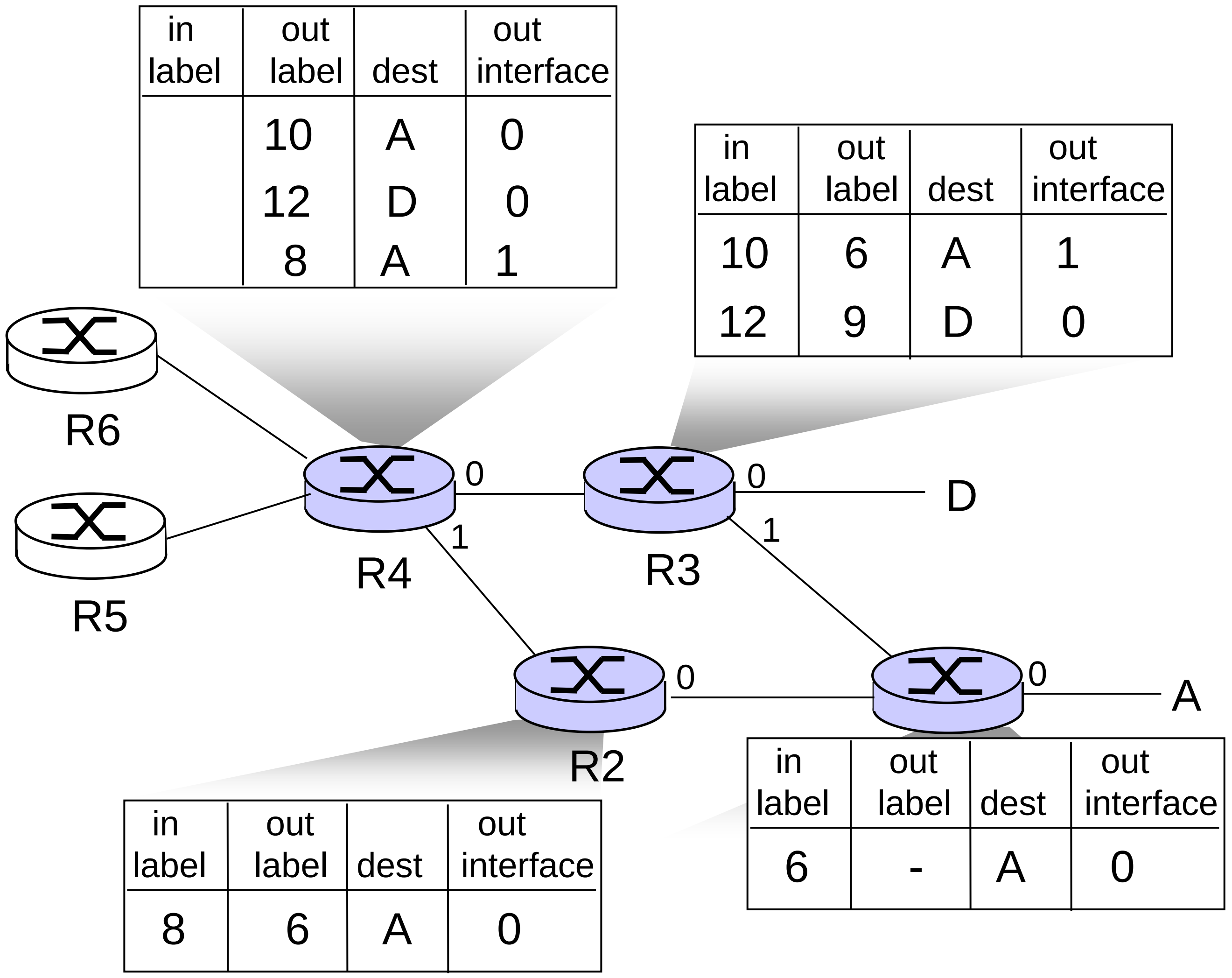
- initial goal: speed up IP forwarding by using fixed length label (instead of IP address) to do forwarding
  - borrowing ideas from Virtual Circuit (VC) approach
  - but IP datagram still keeps IP address!



MPLS capable routers

- a.k.a. label-switched router
- forwards packets to outgoing interface based only on label value (don't inspect IP address)
  - MPLS forwarding table distinct from IP forwarding tables
- signaling protocol needed to set up forwarding
  - RSVP-TE
  - forwarding possible along paths that IP alone would not allow (e.g., source-specific routing) !!
  - use MPLS for traffic engineering
- must co-exist with IP-only routers

MPLS forwarding tables



R4:

in label	out label	dest	out interface
	10	A	0
	12	D	0
	8	A	1

R3:

in label	out label	dest	out interface
10	6	A	1
12	9	D	0

R2:

in label	out label	dest	out interface
8	6	A	0

in label	out label	dest	out interface
6	-	A	0

Link Layer and LANS: 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
  - PPP
  - MPLS
- Next Stop: Wireless and Mobile Networking