# Chapter 5: The Data Link Layer last revised 24/11/03

## Our goals:

❑ understand principles behind data link layer services:

  ❍ error detection, correction

  ❍ sharing a broadcast channel: multiple access

  ❍ link layer addressing

  ❍ reliable data transfer, flow control: *done!*

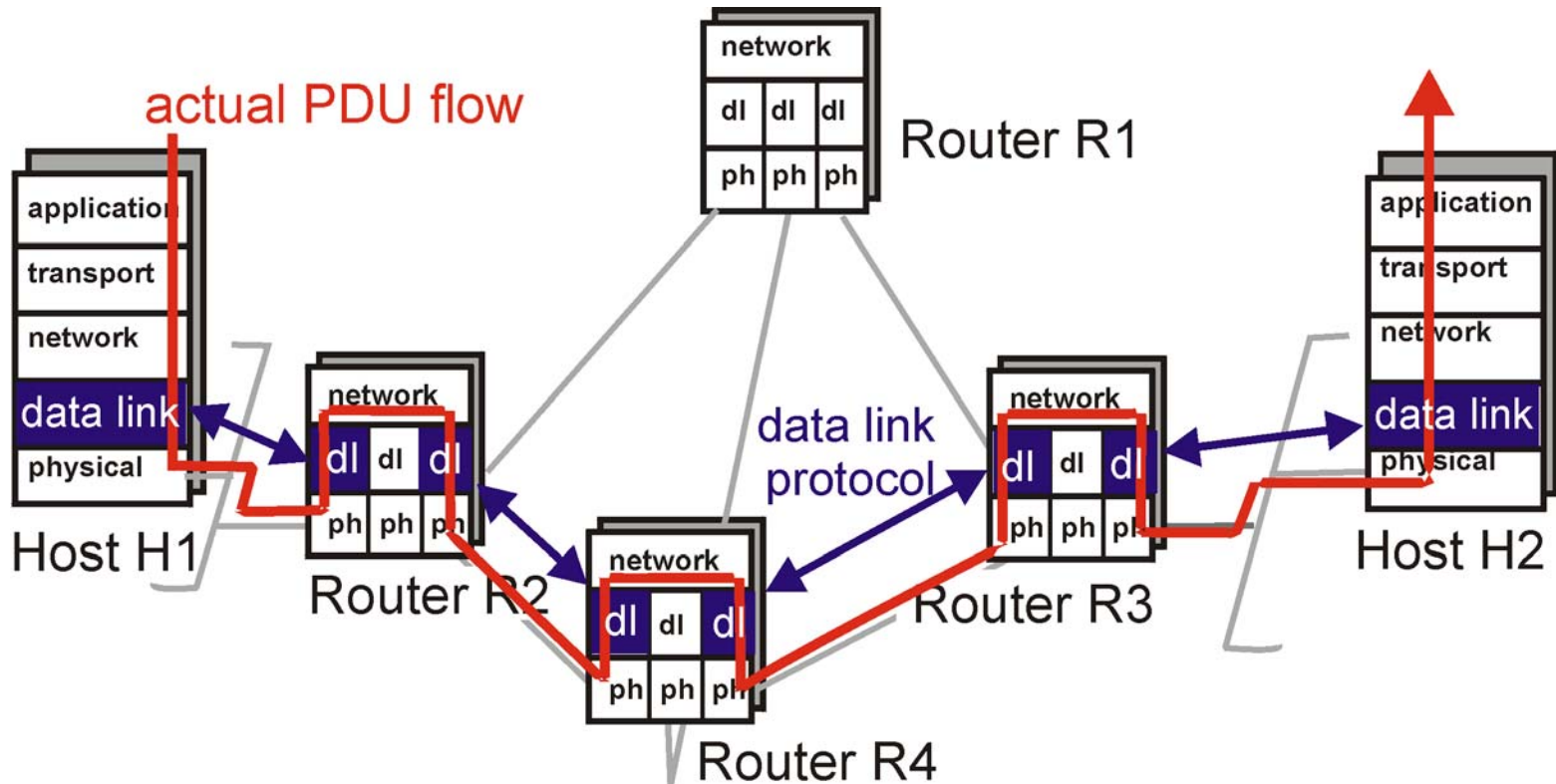❑ instantiation and implementation of various link layer technologies

## Overview:

❑ link layer services

❑ error detection, correction

❑ multiple access protocols and LANs

❑ link layer addressing, ARP

❑ specific link layer technologies:

  ❍ Ethernet
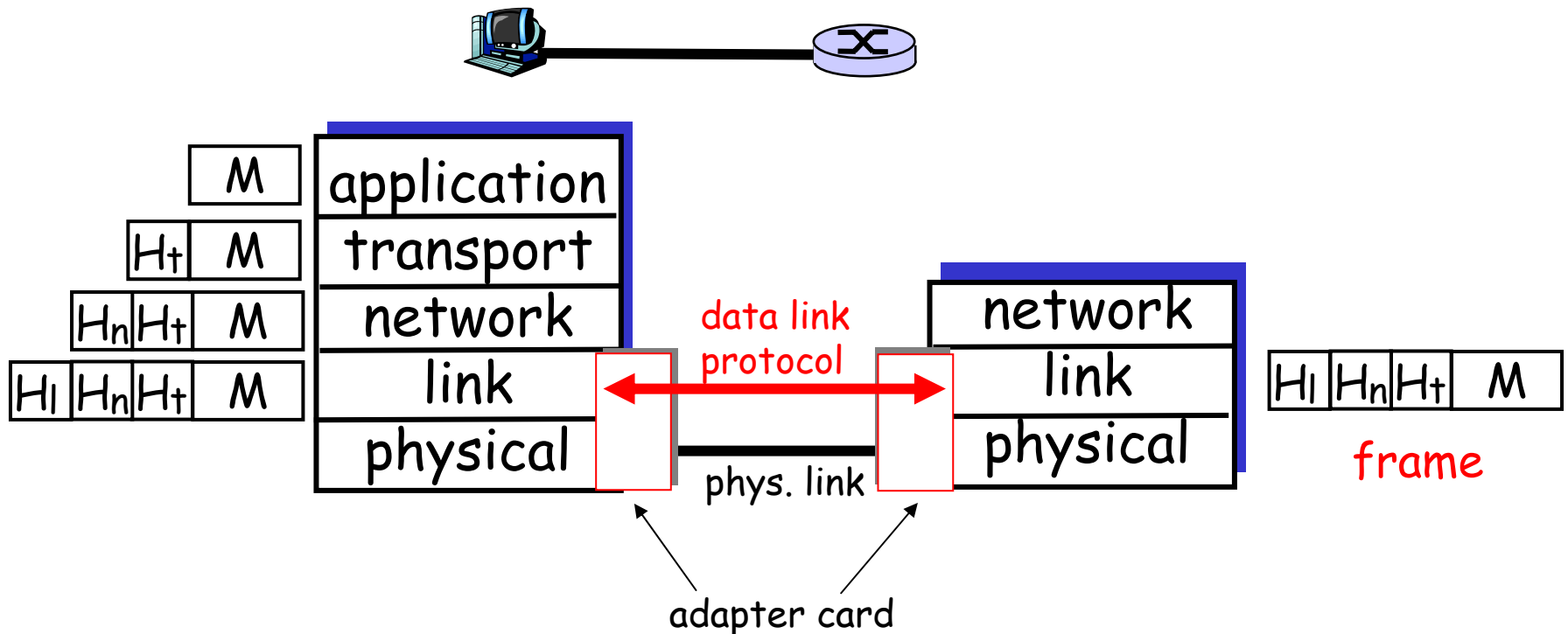
  ❍ hubs, bridges, switches

  ❍ ATM

# Chapter 5 outline

# Link Layer: setting the context

# Link Layer: setting the context

- two *physically connected* devices:
  - host-router, router-router, host-host
- unit of data: *frame*

# Link Layer: setting the context

## Some terminology:

- hosts and routers are **nodes** (bridges and switches too)
- communication channels that connect adjacent nodes along communication path are **links**
  - wired links
  - wireless links
  - LANs
- 2-PDU is a **frame**, encapsulates datagram
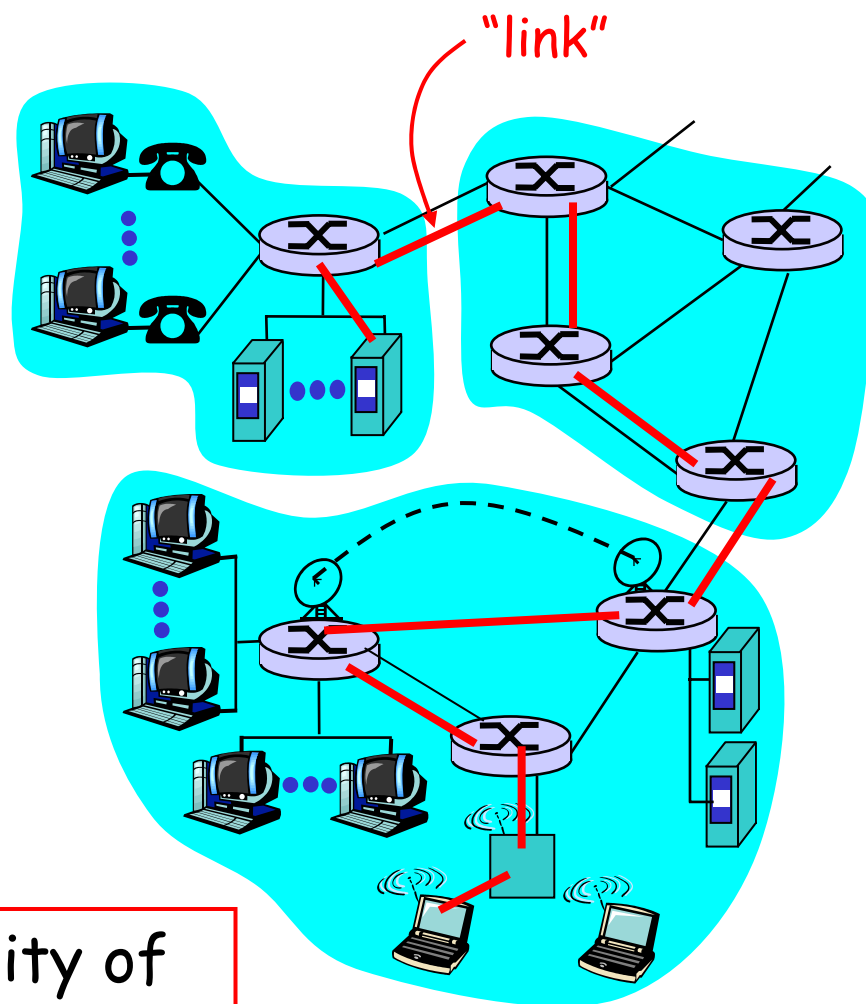
"link"



**data-link layer** 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 on last link
- Each link protocol provides different services
  - e.g., may or may not provide rdt over link

## transportation analogy

- trip from Princeton to Lausanne
  - limo: Princeton to JFK
  - plane: JFK to Geneva
  - train: Geneva to Lausanne
- tourist = datagram
- transport segment = communication link
- transportation mode = link layer protocol
- travel agent = routing algorithm

# Link Layer Services

❒ Framing, link access:
  ○ encapsulate datagram into frame, adding header, trailer
  ○ channel access if shared medium
  ○ 'physical 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:*
- ○ pacing between adjacent sending and receiving nodes

□ *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

datagram

sending
node

link layer protocol

rcving
node

frame

adapter

frame

adapter
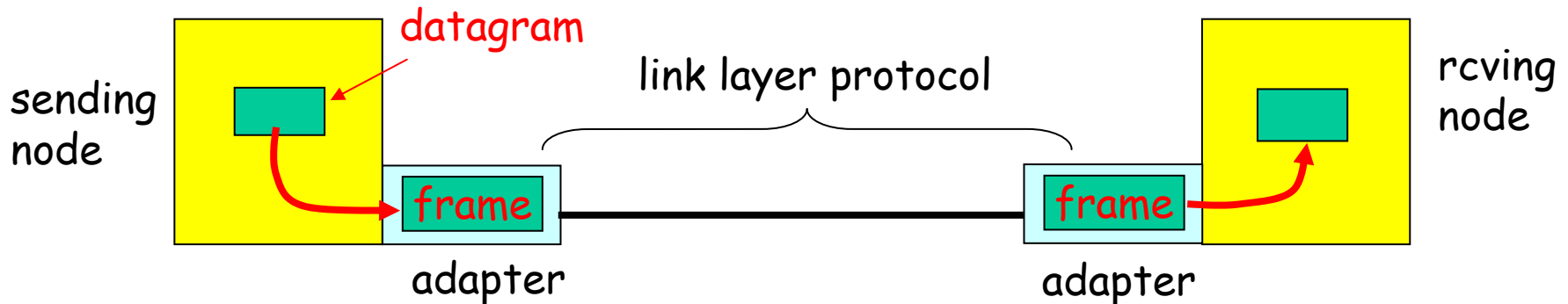
- link layer implemented in "adaptor" (aka NIC)
  - Ethernet card, PCMCIA card, 802.11 card
  - typically includes: RAM, DSP chips, host bus interface, and link interface
- sending side:
  - encapsulates datagram in a frame
  - adds error checking bits, rdt, flow control, etc.

- receiving side
  - looks for errors, rdt, flow control, etc
  - extracts datagram, passes to rcving node
- adapter is semi-autonomous
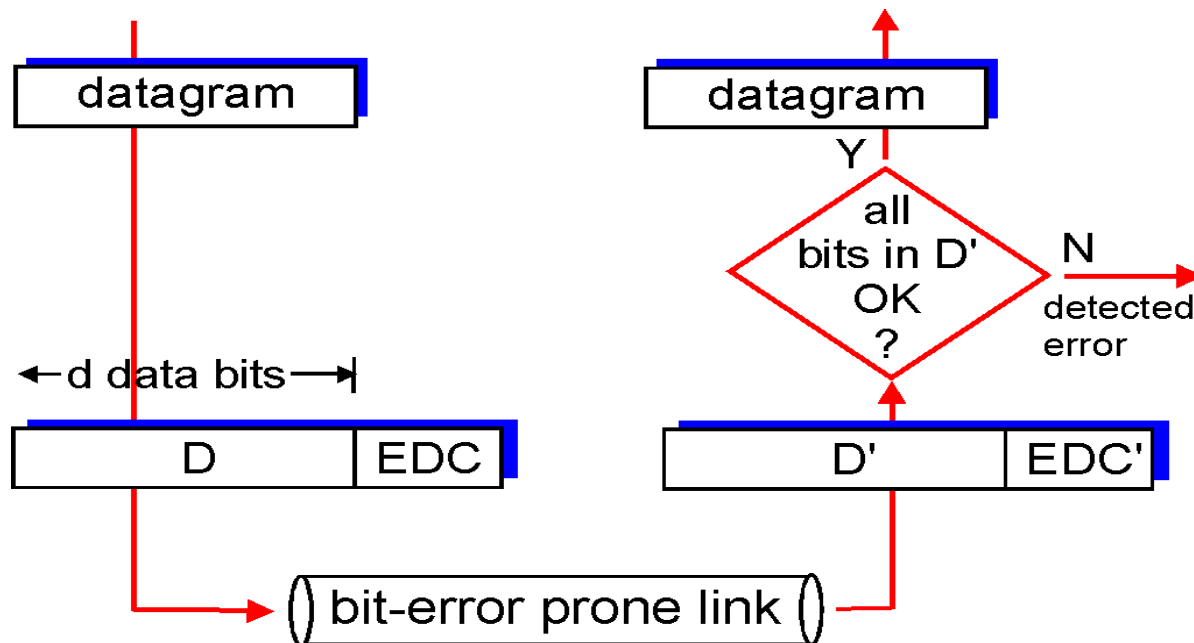- link & physical layers

# Chapter 5 outline

# Error Detection

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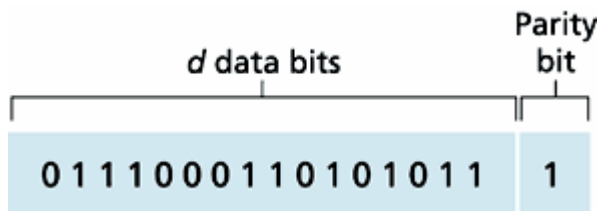D    = Data protected by error checking, may include header fields

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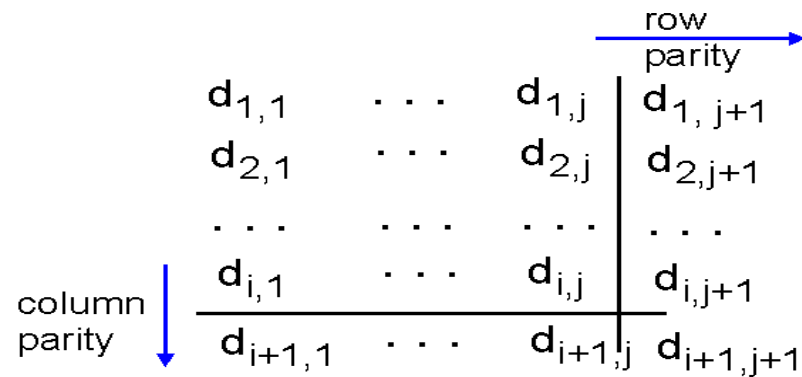  • larger EDC field yields better detection and correction

# Parity Checking

## Single Bit Parity:
**Detect single bit errors**

d data bits | Parity bit

0 1 1 1 0 0 0 1 1 0 1 0 1 0 1 1 | 1

## Two Dimensional Bit Parity:
**Detect *and correct* single bit errors**

row parity →

$$d_{1,1} \quad \cdots \quad d_{1,j} \mid d_{1,\,j+1}$$
$$d_{2,1} \quad \cdots \quad d_{2,j} \mid d_{2,j+1}$$
$$\cdots \qquad \cdots \qquad \cdots$$

column parity ↓

$$d_{i,1} \quad \cdots \quad d_{i,j} \mid d_{i,j+1}$$
$$d_{i+1,1} \quad \cdots \quad d_{i+1,j} \mid d_{i+1,j+1}$$

```
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

# Internet checksum

**Goal:** detect "errors" (e.g., flipped bits) in transmitted segment (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?*

# Checksumming: Cyclic Redundancy Check

- view data bits, D, as coefficients of polynomial
- choose r+1 bit pattern (generator), G
- goal: choose r CRC bits, R, such that
  - <D,R> exactly divisible *(polynomial division)* 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 (ATM, HDCL)



$$D * 2^r \ \text{XOR} \ R$$

*bit pattern*
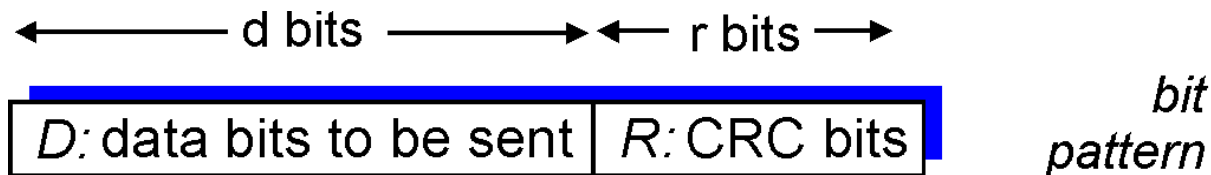
*mathematical formula*

❐ Bits in word are coefficients of polynomial
D = 101110   then   $D(x) = x^5 + x^3 + x^2 + x$

❐ Arithmetic on coefficients is mod 2
$$(x+1)^2 = x^2 + 1$$
In particular   $P(x) + P(x) = 0$

❐ $R(x)$ is the remainder of $P(x)$ divided by $G(x)$ if
$P(x) = A(x)\,G(x) + R(x)$   where   $\deg(R) < \deg(G)$

❐ $G(x)$ divides $P(x) + R(x)$ since
$P(x) + R(x) = A(x)\,G(x) + R(x) + R(x) = A(x)\,G(x)$

❐ $2^r D$ corresponds to   $x^r D(x)$

❐ Given D and G,
  ❍ CRC finds $R(x)$,  remainder when $x^r D(x)$ is divided by $G(x)$.
  ❍ R is word corresponding to $R(x)$.
  ❍ CRC is   $D \cdot 2^r$  XOR  R

# Example (r=3)

D =     101110          $D(x) = x^5 + x^3 + x^2 + x$

G =       1001          $G(x) = x^3 + 1$

$2^r$D = 101110000      $x^3 D(x) = x^8 + x^6 + x^5 + x^4$

---

$x^3 D(x) = G(x)(x^5 + x^3 + x + 1) + (x+1)$

So  $R(x) = x+1$  and  R = 011

---

CRC will transmit

D·$2^r$ XOR R  = 101110011

# Chapter 5 outline

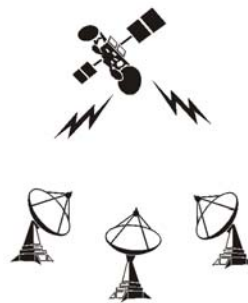# Multiple Access Links and Protocols

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
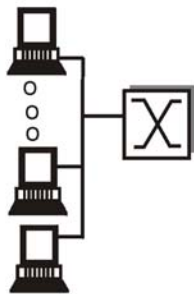
shared wire
(e.g. Ethernet)

shared wireless
(e.g. Wavelan)

satellite

Blah, blah, blah

ZZZzzzzzzzzzz

cocktail party

5: DataLink Layer

18

# Multiple Access protocols

❑ single shared communication channel

❑ two or more simultaneous transmissions by nodes: interference
  ❍ only one node can send successfully at a time

❑ *multiple access protocol:*
  ❍ distributed algorithm that determines how stations share channel, i.e., determine when station can transmit
  ❍ communication about channel sharing must use channel itself!
  ❍ what to look for in multiple access protocols:
    • synchronous or asynchronous
    • information needed about other stations
    • robustness (e.g., to channel errors)
    • performance

# 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

# MAC Protocols: a taxonomy

Three broad classes:

□ Channel Partitioning    TDMA, FDMA, CDMA
  ○ divide channel into smaller "pieces" (time slots, frequency)
  ○ allocate piece to node for exclusive use

□ Random Access    ALOHA, CSMA, CSMA/CD, CSMA/CA
  ○ allow collisions
  ○ "recover" from collisions

□ "Taking turns"    Polling, Token passing
  ○ tightly coordinate shared access to avoid collisions

### Goal: efficient, fair, simple, decentralized

# 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
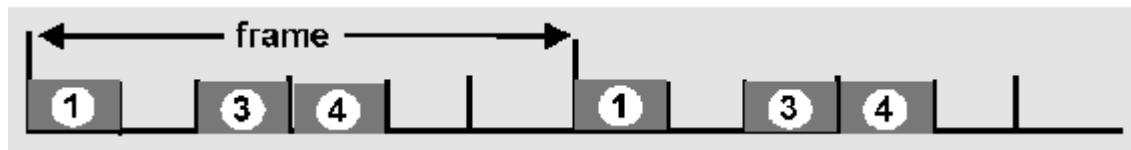
# Channel Partitioning MAC protocols: FDMA

## 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
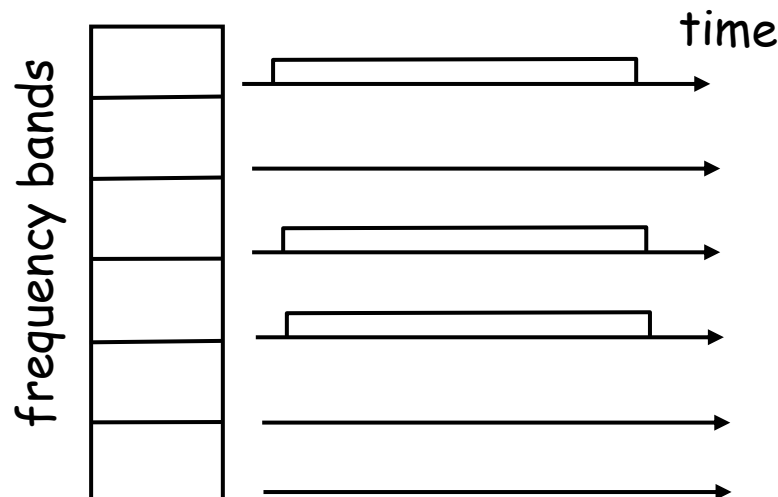
# Channel Partitioning (CDMA)

CDMA (Code Division Multiple Access)

- unique "code" assigned to each user; i.e., code set partitioning
- used mostly in wireless broadcast channels (cellular, satellite, etc)
- all users share same frequency, but each user has own "chipping" sequence (i.e., code) to encode data
- *encoded signal* = (original data) X (chipping sequence)
- *decoding:* inner-product of encoded signal and chipping sequence
- allows multiple users to "coexist" and transmit simultaneously with minimal interference (if codes are "orthogonal")

# CDMA Encode/Decode
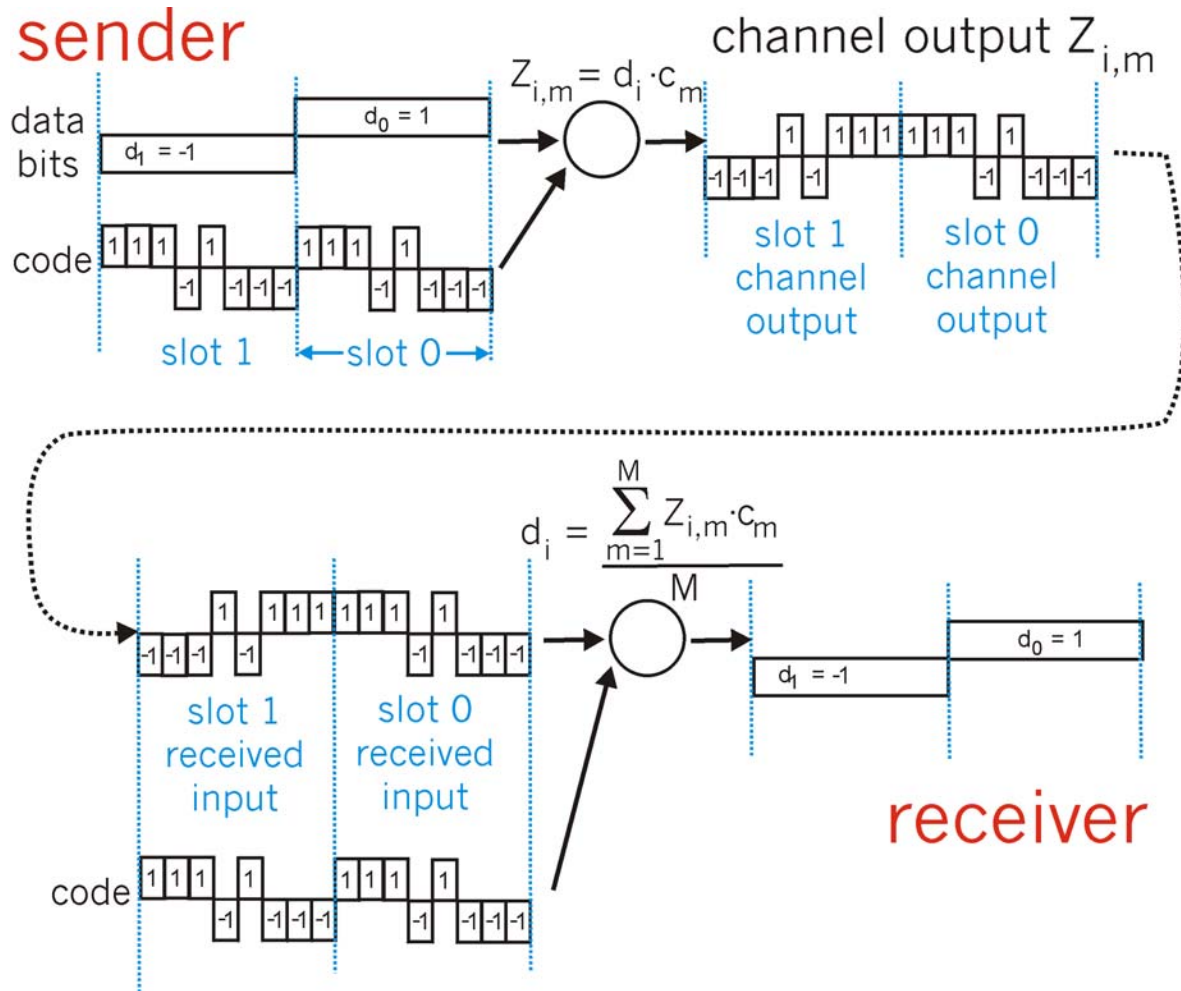
# CDMA: two-sender interference
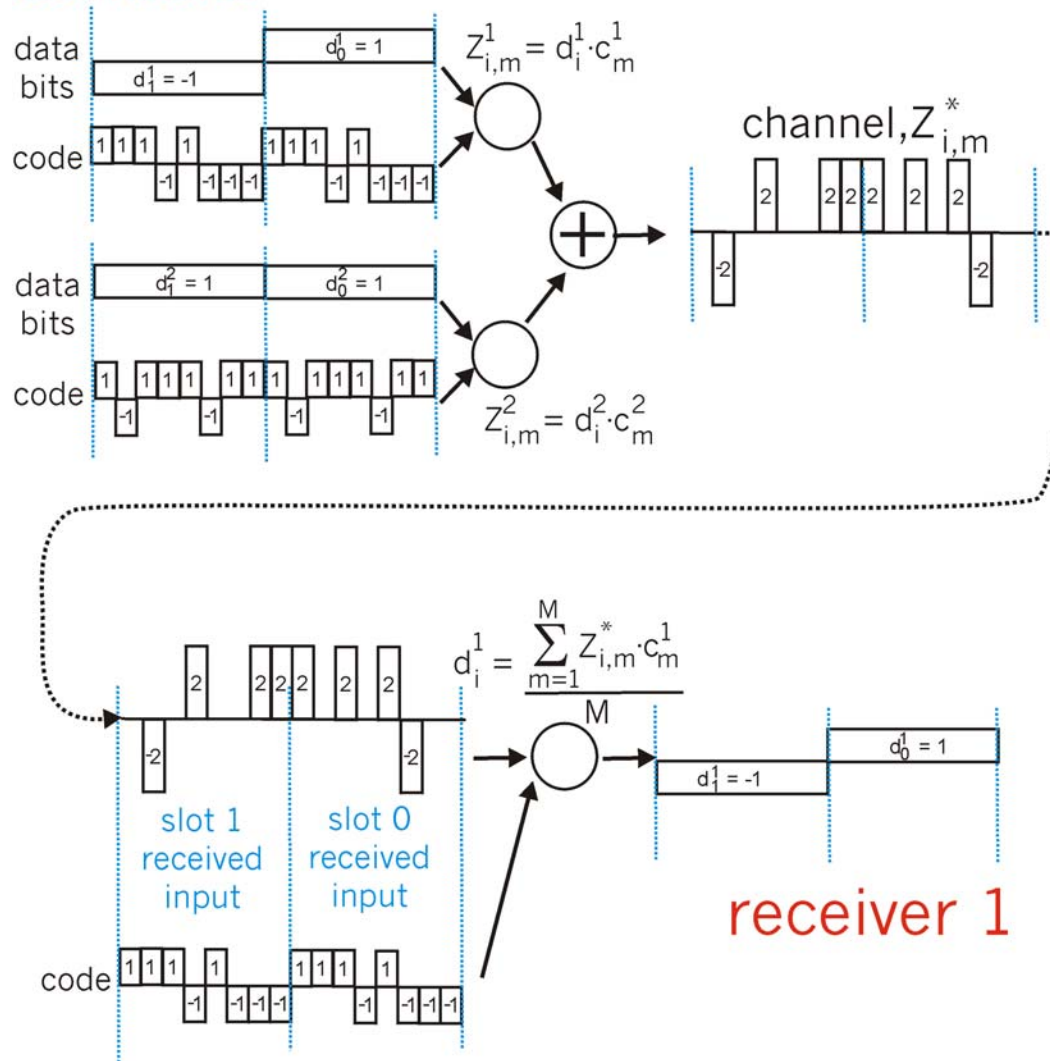
# Random Access Protocols

□ 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

- time is divided into equal size slots (= pkt trans. time)
- node with new arriving pkt: transmit at beginning of next slot
- if collision: retransmit pkt in future slots with probability p, until successful.
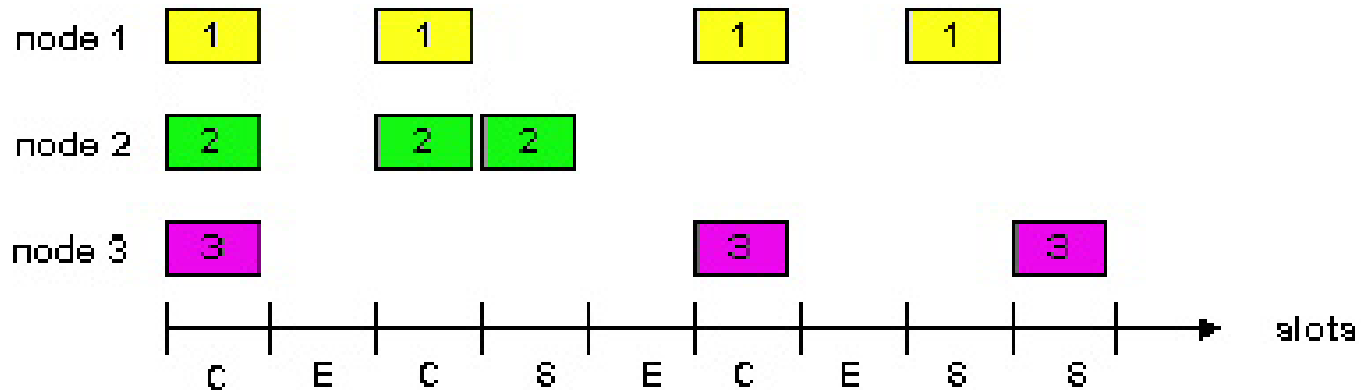


Success (S), Collision (C), Empty (E) slots

# 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
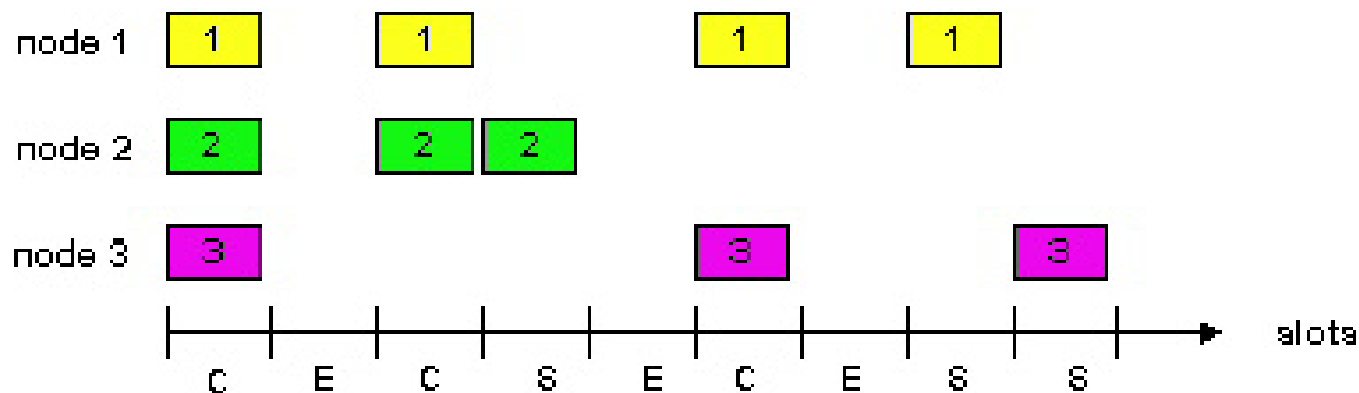


## Pros

❒ 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

# Slotted Aloha efficiency

**Efficiency** is the long-run fraction of successful slots when there's many nodes, each with many frames to send

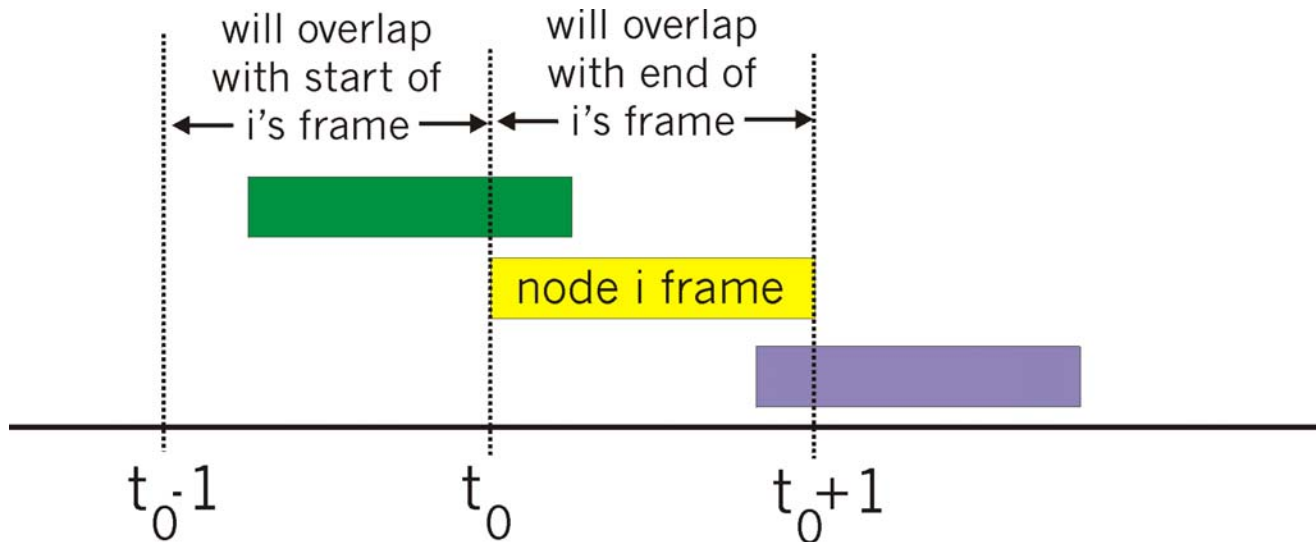- Suppose N nodes with many frames to send, each transmits in slot with probability $p$
- prob that 1st node has success in a slot = $p(1-p)^{N-1}$
- prob that any node has a success = $Np(1-p)^{N-1}$

- 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 (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 $[p_0-1,p_0]$ ·

   P(no other node transmits in $[p_0-1,p_0]$

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

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

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

= 1/(2e) = .18

Even worse !

# CSMA:  (Carrier Sense Multiple Access)

**CSMA**: listen before transmit:

- ☐ If channel sensed idle: transmit entire pkt
    - ○ If collision occurs has to retransmit again
- ☐ If channel sensed busy, defer transmission
    - ○ P-Persistent CSMA: (for slotted channels) retry immediately with probability p when channel becomes idle (may cause instability)
    - ○ Non-persistent CSMA: (for nonslotted channels) retry after random interval
- ☐ human analogy: don't interrupt others!

# CSMA collisions

spatial layout of nodes along ethernet



collisions *can* occur:

propagation delay means two nodes may not hear each other' transmission

collision:

entire packet transmission time wasted

note:

role of distance and propagation delay in determining collision prob.
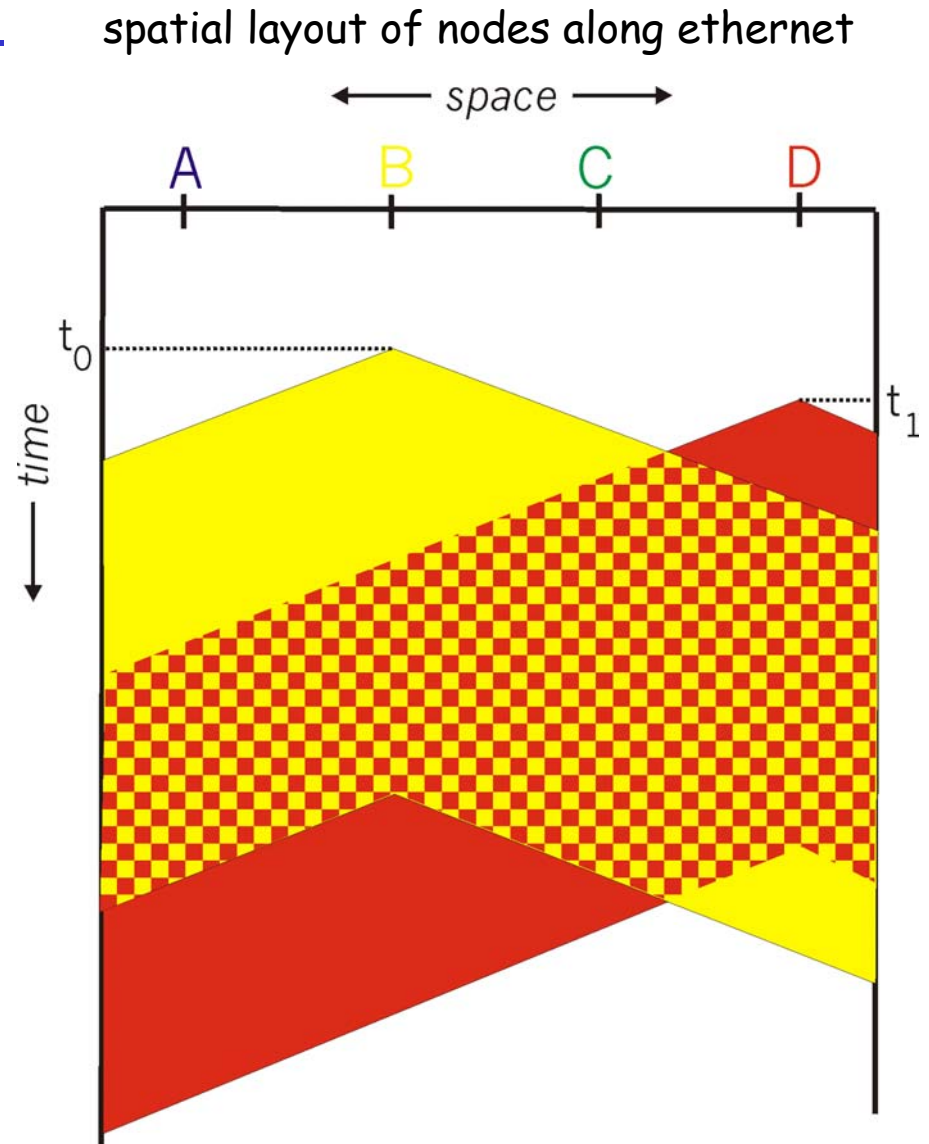
# CSMA/CD (Collision Detection)

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

□ human analogy: the polite conversationalist

# CSMA/CD collision detection
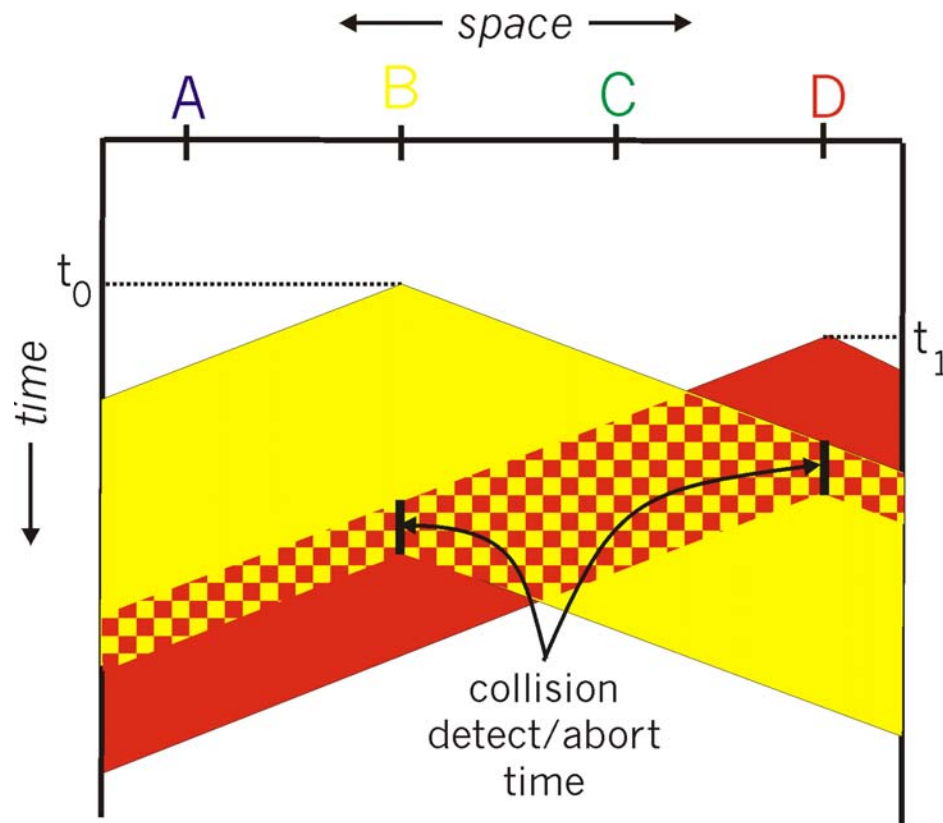
# "Taking Turns" MAC protocols

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

"taking turns" protocols

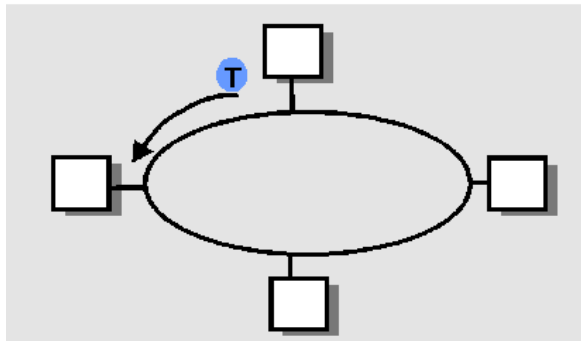  look for best of both worlds!

# "Taking Turns" MAC protocols

**Polling:**

- master node "invites" slave nodes to transmit in turn

- concerns:
  - polling overhead
  - latency
  - single point of failure (master)

**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,Code Division, Frequency Division

○ Random partitioning (dynamic),
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - carrier sensing: easy in some technologies (wire), hard in others (wireless)
  - CSMA/CD used in Ethernet

○ Taking Turns
  - polling from a central site, token passing

# Chapter 5 outline

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 LAN addresses and ARP
- 5.5 Ethernet

- 5.6 Hubs, bridges, and switches
- 5.7 Wireless links and LANs
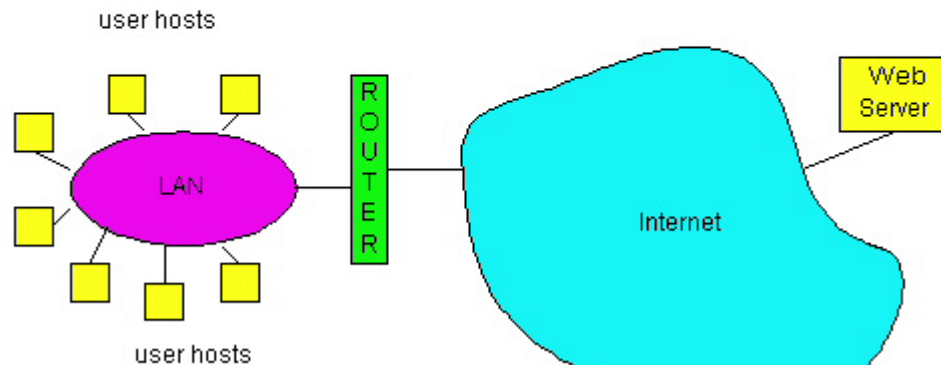- 5.8 PPP
- 5.9 ATM
- 5.10 Frame Relay

# LAN technologies

Data link layer so far:

- services, error detection/correction, multiple access

Next: LAN technologies

- addressing
- Ethernet
- hubs, bridges, switches
- 802.11
- PPP
- ATM

# LAN Addresses and ARP

32-bit IP address:

❐ *network-layer* address

❐ used to get datagram to destination IP network
(recall IP network definition)

LAN (or MAC or physical or Ethernet) address:

❐ used to get datagram from one interface to another
physically-connected interface (same network)

❐ 48 bit MAC address (for most LANs)
burned in the adapter ROM

# LAN Addresses and ARP

Each adapter on LAN has unique LAN address



= adapter

node — 1A-23-F9-CD-06-9B

node — 5C-66-AB-90-75-B1

LAN

node — 88-B2-2F-54-1A-0F

node — 49-BD-D2-C7-56-2A
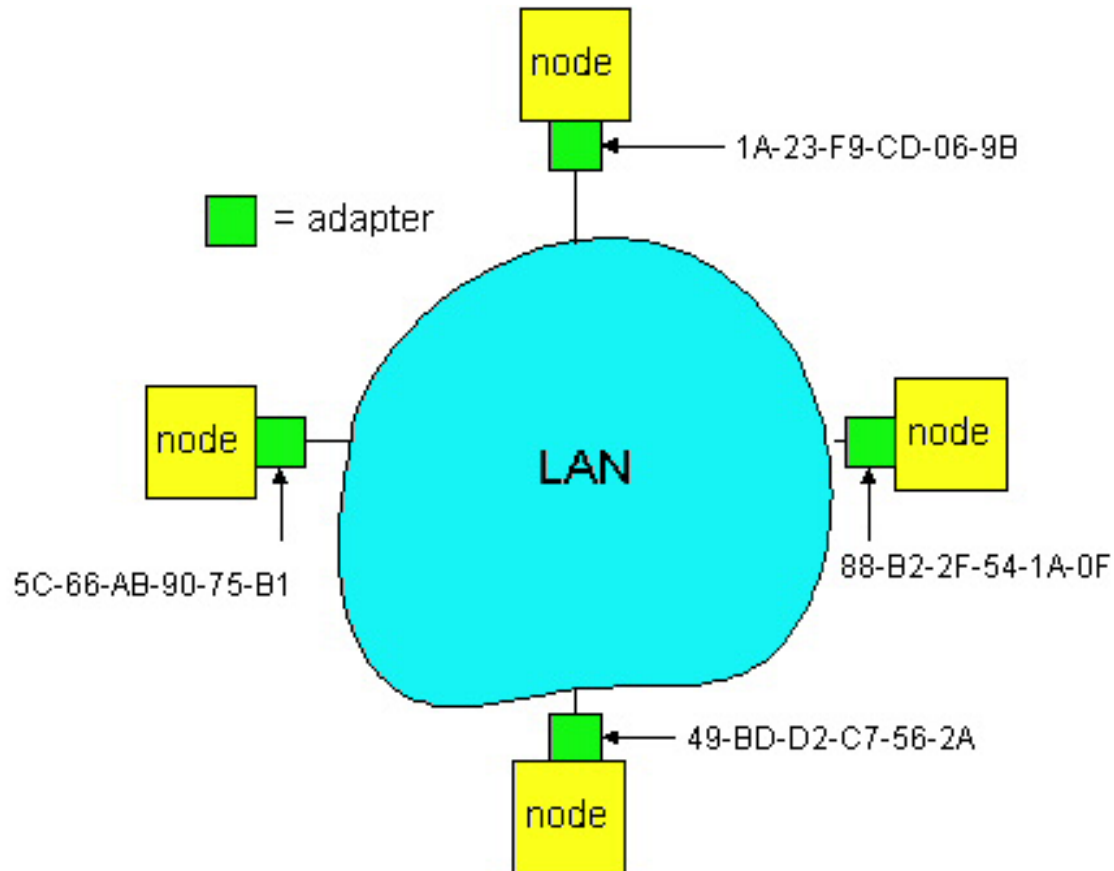
# LAN Address (more)

❒ MAC address allocation administered by IEEE
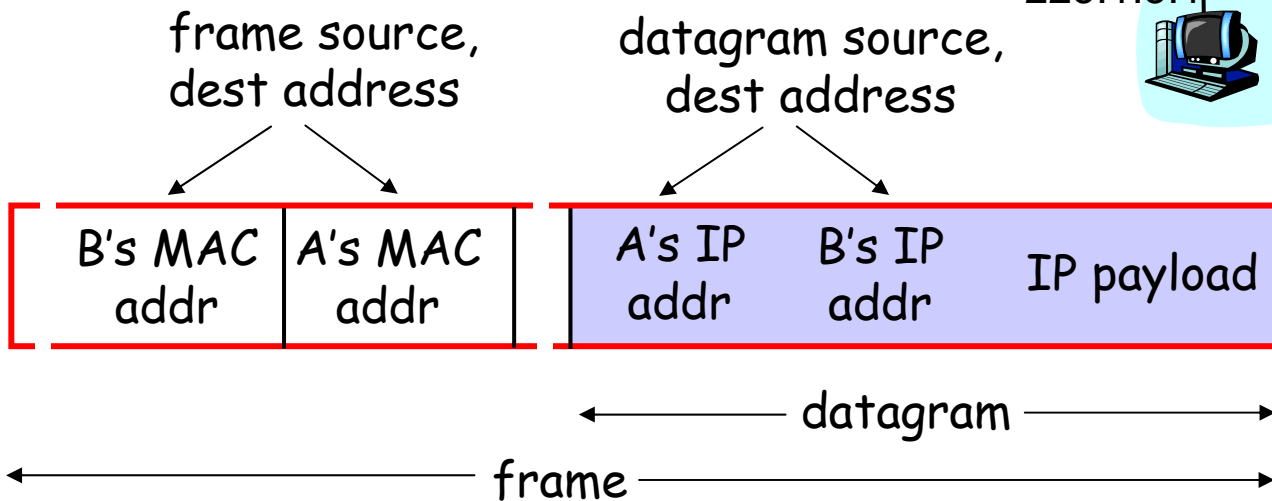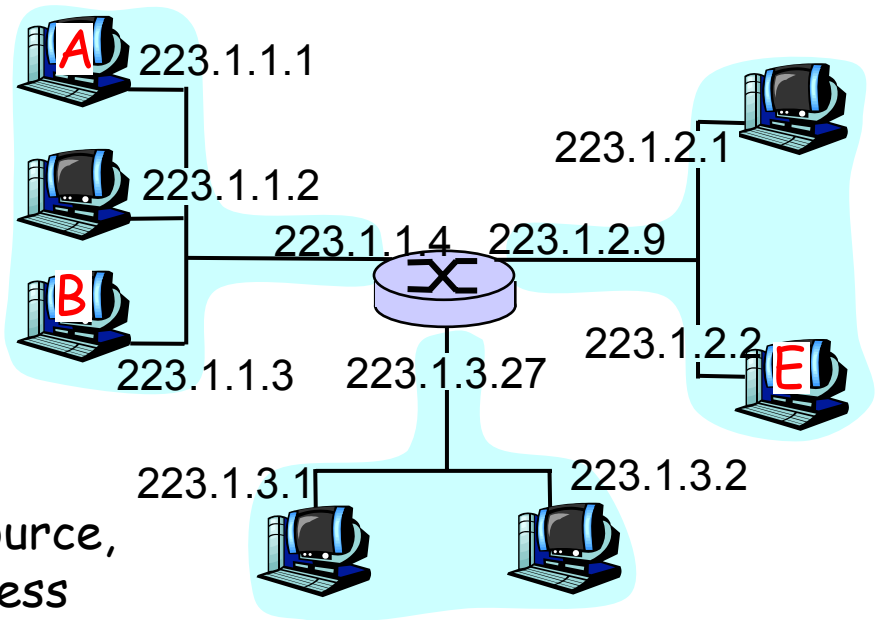
❒ manufacturer buys portion of MAC address space (to assure uniqueness)

❒ Analogy:

　　　(a) MAC address: like Social Security Number

　　　(b) IP address: like postal address

❒ MAC flat address  => portability

　　○ can move LAN card from one LAN to another

❒ IP hierarchical address NOT portable

　　○　depends on IP network to which node is attached

# Recall earlier routing discussion

Starting at A, given IP datagram addressed to B:

- look up net. address of B, find B on same net. as A
- link layer send datagram to B inside link-layer frame

223.1.1.1
223.1.1.2
223.1.1.4   223.1.2.9
223.1.1.3   223.1.3.27
223.1.2.1
223.1.2.2
223.1.3.1   223.1.3.2

| frame source, dest address | | | datagram source, dest address | | |
|---|---|---|---|---|---|
| B's MAC addr | A's MAC addr | | A's IP addr | B's IP addr | IP payload |

datagram
frame

# ARP: Address Resolution Protocol

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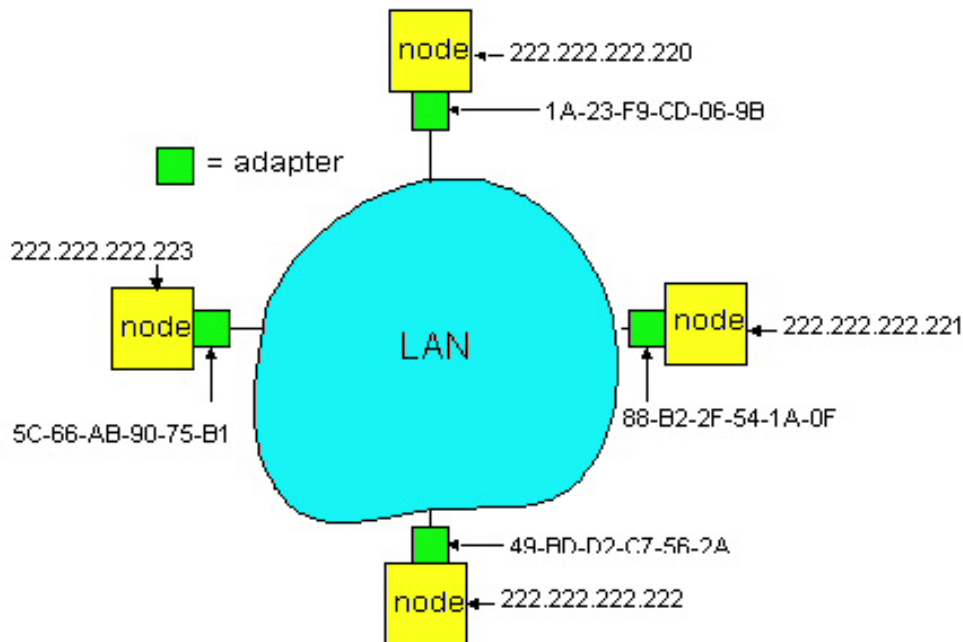


□ Each IP node (Host, Router) on LAN has ARP table

□ 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

- A wants to send datagram to B, and A knows B's IP address.
- Suppose B's MAC address is not in A's ARP table.
- A broadcasts ARP query packet, containing B's IP address
  - 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)
  - 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

# Routing to another LAN

walkthrough: send datagram from A to B via R
assume  A knows B IP address
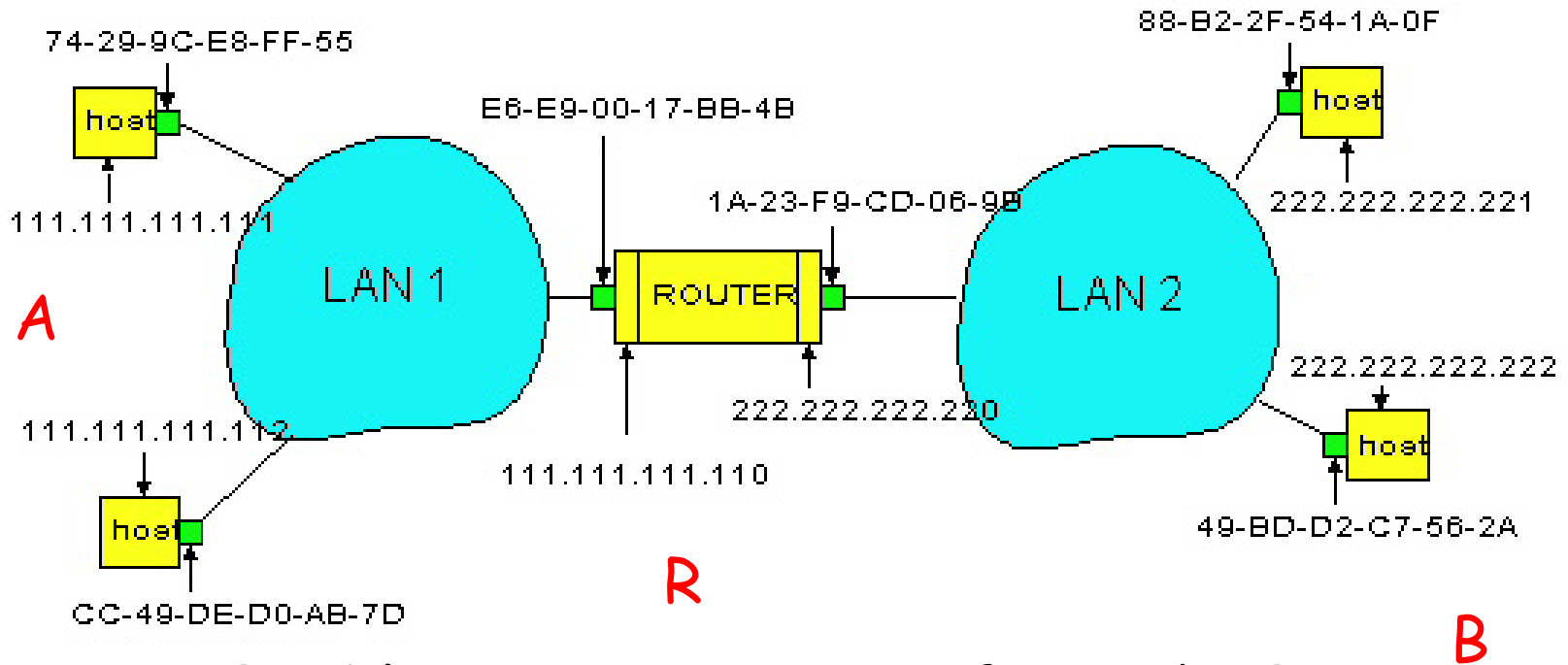


□ Two ARP tables in  router R, one for each IP
network (LAN)

- A creates 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 data link layer sends frame
- R's data link layer receives frame
- R removes IP datagram from Ethernet frame, sees its destined to B
- R uses ARP to get B's physical layer address
- R creates frame containing A-to-B IP datagram sends to B
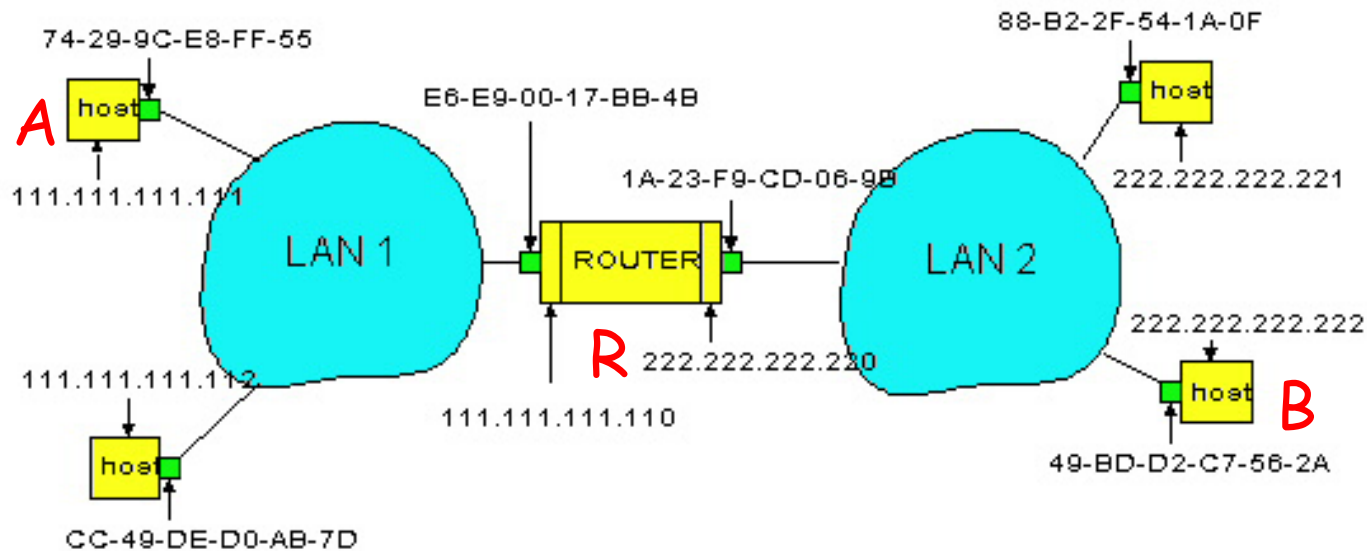
# Ethernet

"dominant" LAN technology:

- cheap $20 for 100Mbs!
- first widely used LAN technology
- Simpler, cheaper than token LANs and ATM
- Kept up with speed race: 10, 100, 1000 Mbps
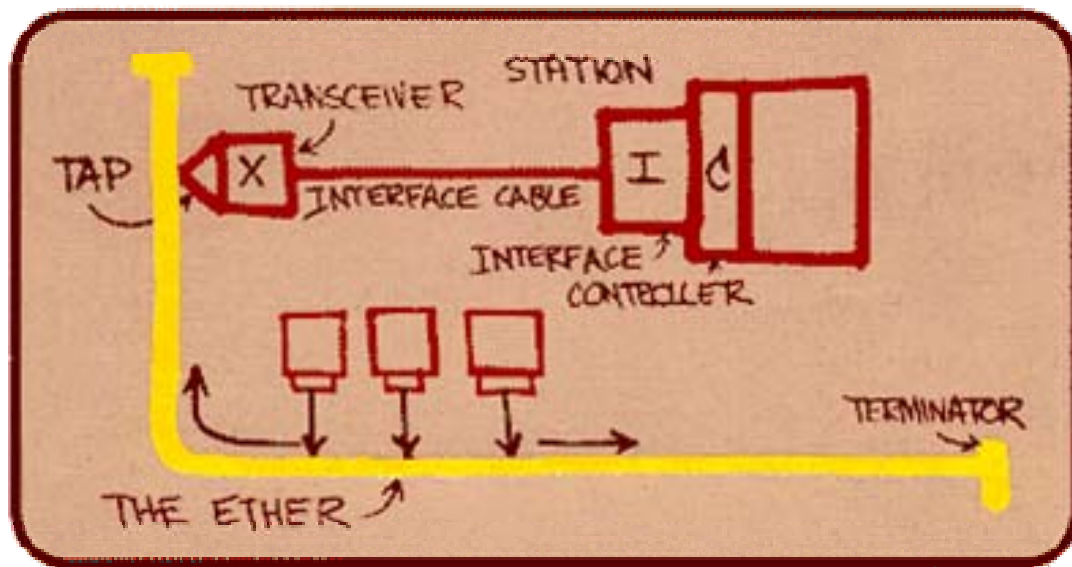


Metcalfe's Ethernet sketch

# Ethernet Frame Structure

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

| Preamble | Dest. Address | Source Address | | Data | CRC |
|---|---|---|---|---|---|

Type
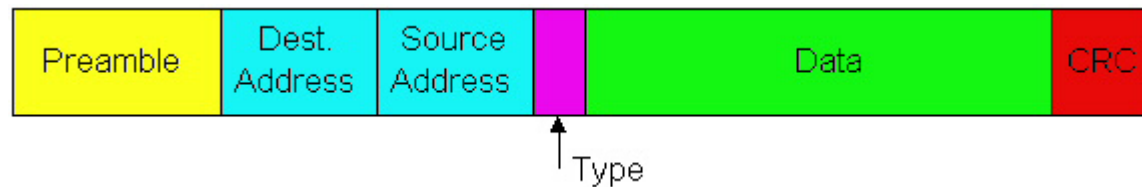
Preamble:

❑ 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 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



Preamble | Dest. Address | Source Address | | Data | CRC

Type
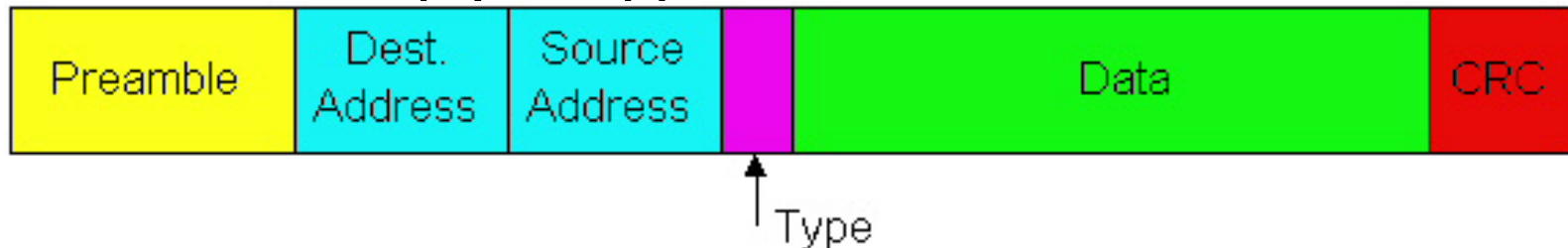
# Unreliable, connectionless service

□ 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

    *random time* depends upon # collisions so far

# Ethernet CSMA/CD algorithm

1. Adaptor gets datagram from link and 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 $\{0,1,2,…,2^m-1\}$. Adapter waits K*512 bit times and returns to Step 2

# Ethernet: slotless, uses CSMA/CD

create frame from datagram

**A**: sense channel; **if** idle

  **then** {

        transmit frame and monitor the channel;

        **If** detect another transmission

         **then** {

           abort and send jam signal;

           update # collisions;

           delay as required by exponential backoff algorithm;

           goto A

         }

       **else** {done with the frame; set collisions to zero}

    }

  **else** {wait until ongoing transmission is over and goto A}

# Ethernet's CSMA/CD (more)

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

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   (btt = .1 microsec for 10Mbps Ethernet)

□ after second collision: choose K from {0,1,2,3}…

□ after ten or more collisions, choose K from {0,1,2,3,4,…,1023}

# CSMA/CD efficiency

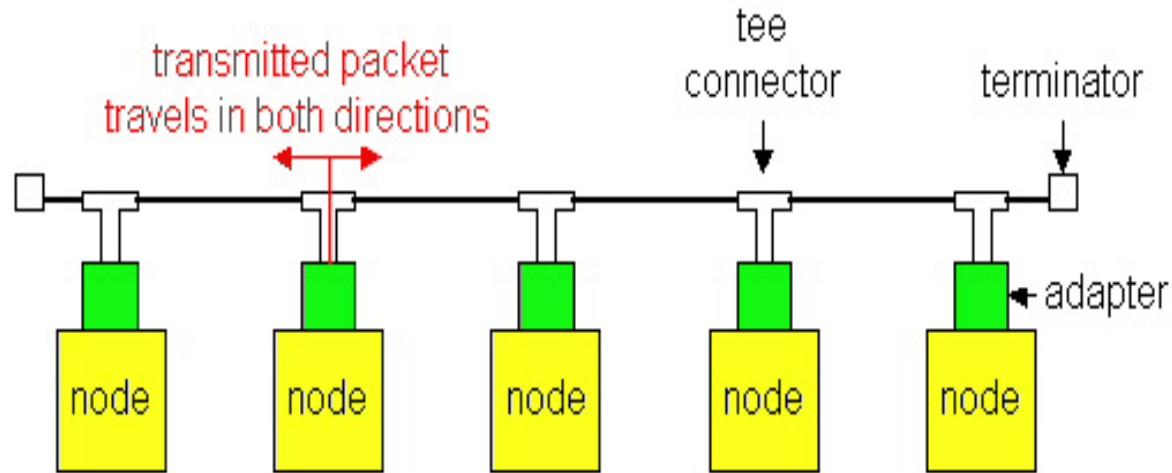□ $T_{prop}$ = max prop between 2 nodes in LAN
□ $t_{trans}$ = time to transmit max-size frame

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

□ 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

# Ethernet Technologies: 10Base2

- 10: 10Mbps; 2: under 200 meters max cable length
- thin coaxial cable in a bus topology



- repeaters used to connect up to 5 multiple segments
- repeater repeats bits it hears on one interface to its other interfaces: physical layer device only!
- has become a legacy technology

# 10BaseT and 100BaseT

- ❐ 10/100 Mbps rate; latter called "fast ethernet"
- ❐ T stands for Twisted Pair
- ❐ Nodes connect to a hub: "star topology"; 100 m max distance between nodes and hub
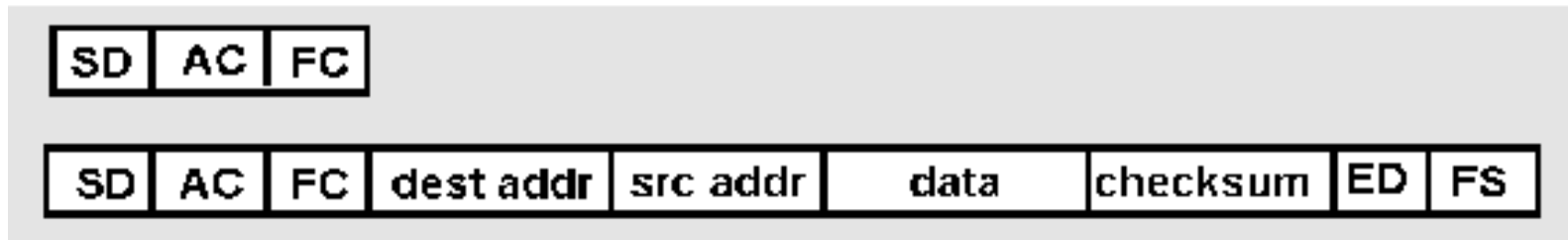
nodes

hub
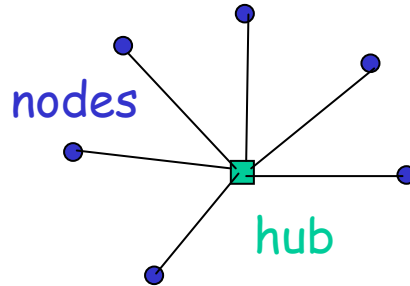
- ❐ Hubs are essentially physical-layer repeaters:
  - ○ bits coming in one link go out all other links
  - ○ no frame buffering
  - ○ no CSMA/CD at hub: adapters detect collisions
  - ○ provides net management functionality

# Gbit Ethernet

- □ use standard Ethernet frame format
- □ allows for point-to-point links and shared broadcast channels
- □ in shared mode, CSMA/CD is used; short distances between nodes to be efficient
- □ uses hubs, called here "Buffered Distributors"
- □ Full-Duplex at 1 Gbps for point-to-point links
- □ 10 Gbps now !

# Token Passing: IEEE802.5 standard

❑ 4 Mbps
❑ max token holding time: 10 ms, limiting frame length

| SD | AC | FC |
|----|----|----|

| SD | AC | FC | dest addr | src addr | data | checksum | ED | FS |
|----|----|----|----|----|----|----|----|----|

❑ SD, ED mark start, end of packet
❑ AC: access control byte:
  ○ token bit: value 0 means token can be seized, value 1 means data follows FC
  ○ priority bits: priority of packet
  ○ reservation bits: station can write these bits to prevent stations with lower priority packet from seizing token after token becomes free

# Token Passing: IEEE802.5 standard

| SD | AC | FC |
|----|----|----|

| SD | AC | FC | dest addr | src addr | data | checksum | ED | FS |
|----|----|----|-----------|----------|------|----------|----|----|

- **FC:** frame control used for monitoring and maintenance
- **source, destination address:** 48 bit physical address, as in Ethernet
- **data:** packet from network layer
- **checksum:** CRC
- **FS:** frame status: set by dest., read by sender
  - set to indicate destination up, frame copied OK from ring
  - DLC-level ACKing

# Chapter 5 outline

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 LAN addresses and ARP
- 5.5 Ethernet

- 5.6 Hubs, bridges, and switches
- 5.7 Wireless links and LANs
- 5.8 PPP
- 5.9 ATM
- 5.10 Frame Relay

# Interconnecting LANs

Q: Why not just one big LAN?

- Limited amount of supportable traffic: on single LAN, all stations must share bandwidth
- limited length: 802.3 specifies maximum cable length
- large "collision domain" (can collide with many stations)
- limited number of stations: 802.5 have token passing delays at each station

# Interconnecting LAN segments

☐ Hubs

☐ Bridges

☐ Switches

    ○ Remark: switches are essentially multi-port bridges.

    ○ What we say about bridges also holds for switches!

# Hubs

□ Physical Layer devices: essentially repeaters operating at bit levels: repeat received bits on one interface to all other interfaces

□ Hubs can be arranged in a hierarchy (or multi-tier design), with backbone hub at its top
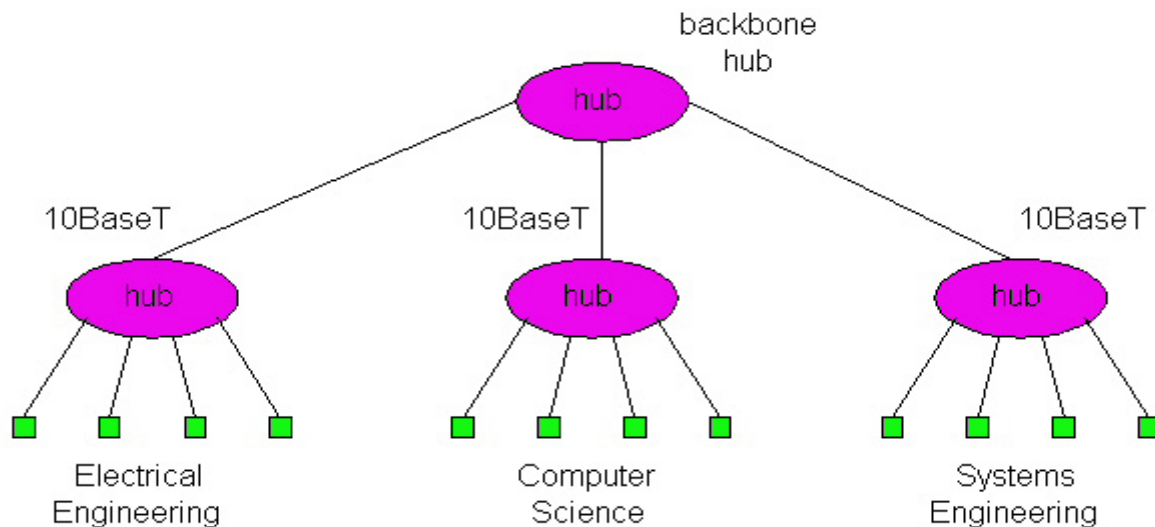
# Hubs (more)

❐ Each connected LAN referred to as LAN **segment**

❐ Hubs <span style="color:red">do not isolate</span> collision domains: node may collide with any node residing at any segment in LAN

❐ Hub Advantages:
- ❍ simple, inexpensive device
- ❍ Multi-tier provides graceful degradation: portions of the LAN continue to operate if one hub malfunctions
- ❍ extends maximum distance between node pairs (100m per Hub)

# Hub limitations

□ single collision domain results in no increase in max throughput

  ○ multi-tier throughput same as single segment throughput

□ individual LAN restrictions pose limits on number of nodes in same collision domain and on total allowed geographical coverage

□ cannot connect different Ethernet types (e.g., 10BaseT and 100baseT)
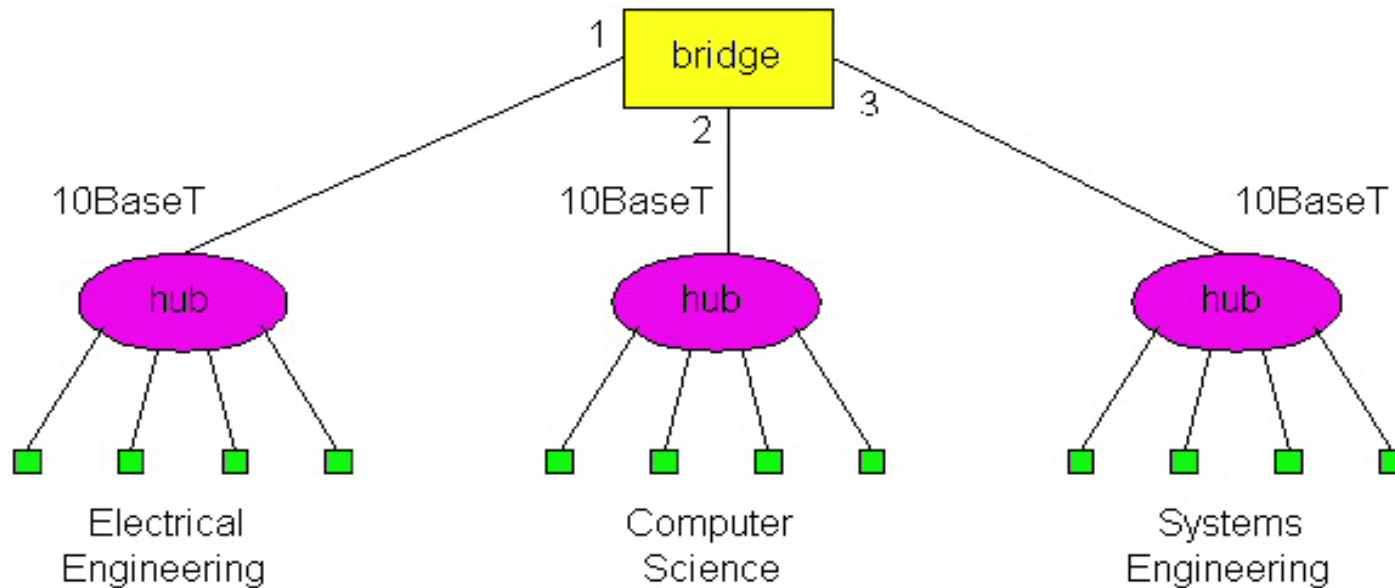
# Bridges

□ **Link layer device**
  ○ stores and forwards Ethernet frames
  ○ examines frame header and selectively forwards frame based on MAC dest address
  ○ when frame is to be forwarded on segment, uses CSMA/CD to access segment

□ transparent
  ○ hosts are unaware of presence of bridges

□ plug-and-play, self-learning
  ○ bridges do not need to be configured

# Bridges: traffic isolation

□ Bridge installation breaks LAN into LAN segments

□ bridges filter packets:

○ same-LAN-segment frames not usually forwarded onto other LAN segments

○ segments become separate collision domains



collision domain

bridge

collision domain

■ = hub

● = host

← LAN segment →    ← LAN segment →

← LAN (IP network) →
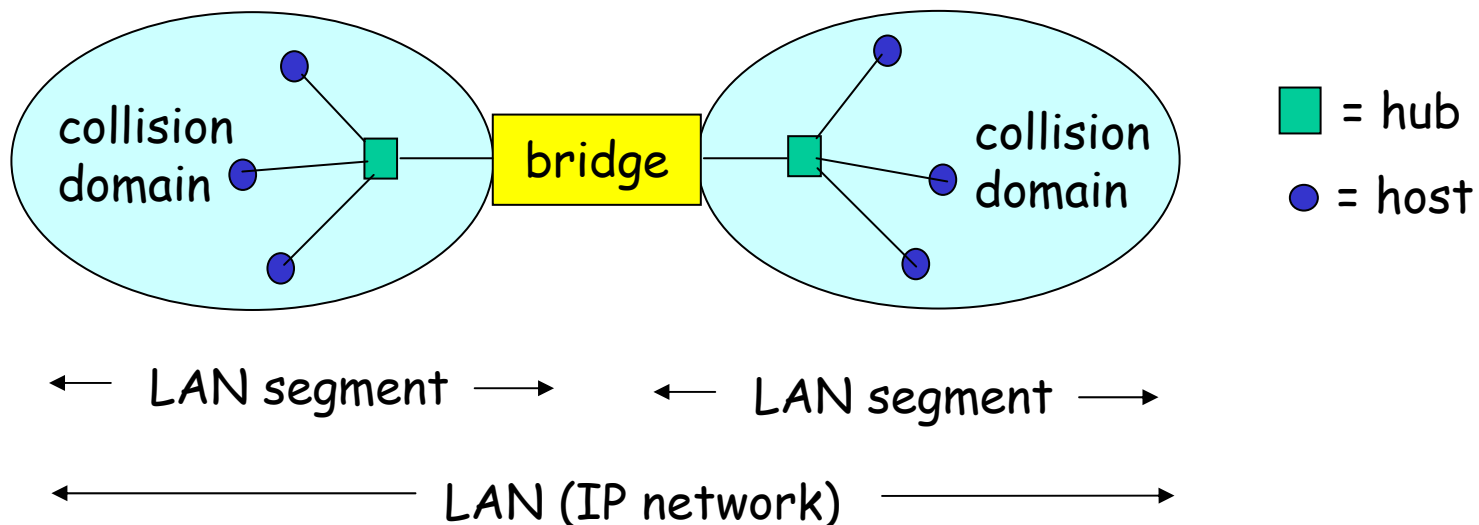
# Bridges: Forwarding



How do determine to which LAN segment to forward frame?
• Looks like a routing problem...

# Bridges: Self learning

❐ A bridge has a bridge table

❐ entry in bridge table:
  ❍ (Node LAN Address, Bridge Interface, Time Stamp)
  ❍ stale entries in table dropped (TTL can be 60 min)

❐ bridges *learn* which hosts can be reached through which interfaces
  ❍ when frame received, bridge "learns" location of sender: incoming LAN segment
  ❍ records sender/location pair in bridge table

# Bridges: Filtering/Forwarding

When bridge receives a frame:

index bridge table using MAC dest address
**if** entry found for destination
  **then{**
    **if** dest on segment from which frame arrived
      **then** drop the frame
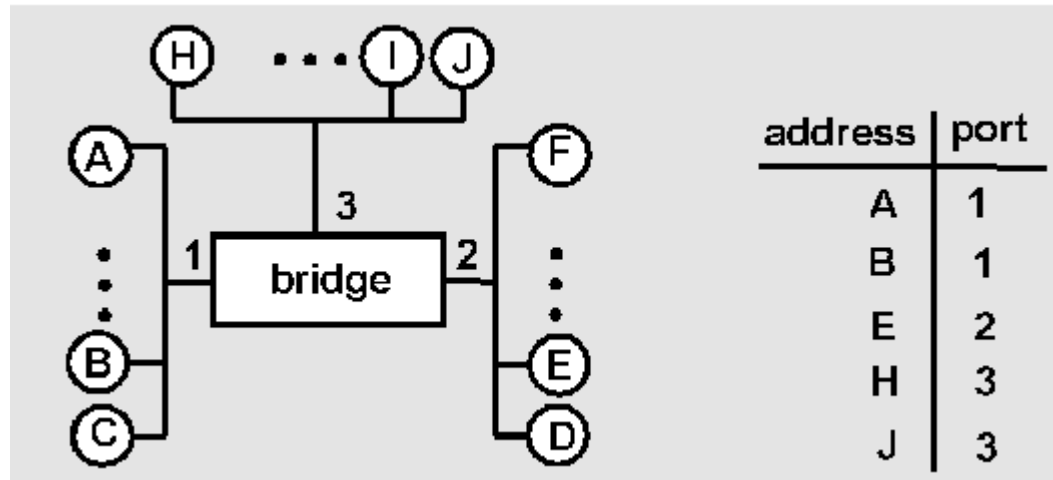      **else** forward the frame on interface indicated
  **}**
  **else** flood

*forward on all but the interface on which the frame arrived*
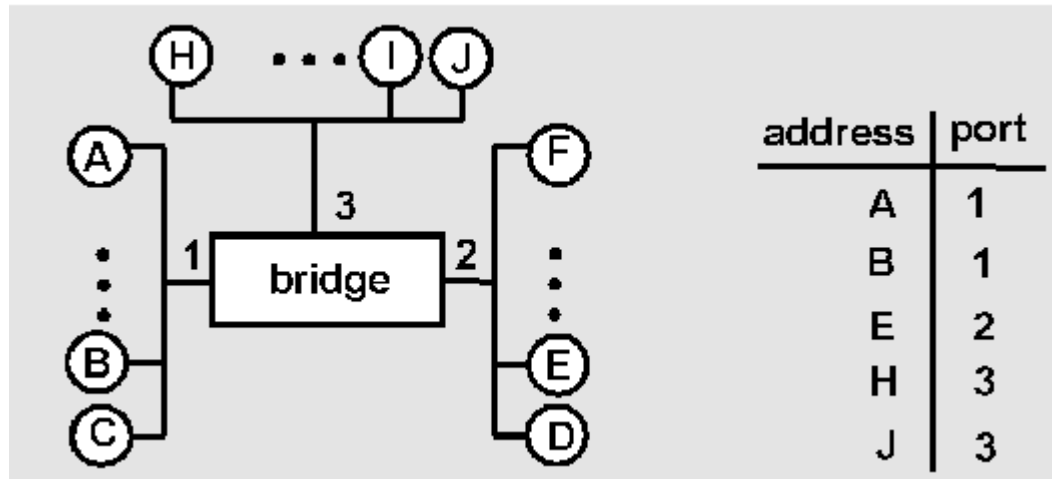
5: DataLink Layer

# Bridge example

Suppose C sends frame to D and D replies back with frame to C.



❑ Bridge receives frame from from C
  ○ notes in bridge table that C is on interface 1
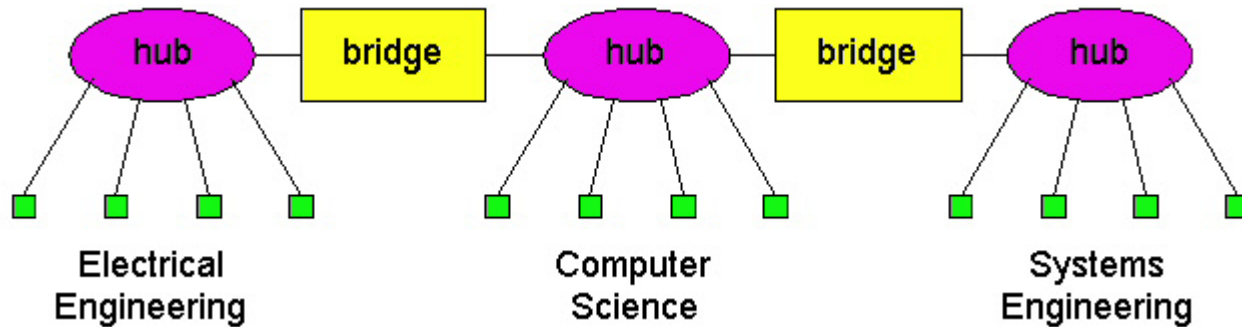  ○ because D is not in table, bridge sends frame into interfaces 2 and 3

❑ frame received by D

# Bridge Learning: example



| address | port |
|---------|------|
| A | 1 |
| B | 1 |
| E | 2 |
| H | 3 |
| J | 3 |

❑ D generates frame for C, sends

❑ bridge receives frame

- ❍ notes in bridge table that D is on interface 2
- ❍ bridge knows C is on interface 1, so *selectively* forwards frame to interface 1

# Bridges: Interconnection without backbone



❒ Not recommended for two reasons:

- single point of failure at Computer Science hub
- all traffic between EE and SE must path over CS segment

# Bridges: Backbone configuration
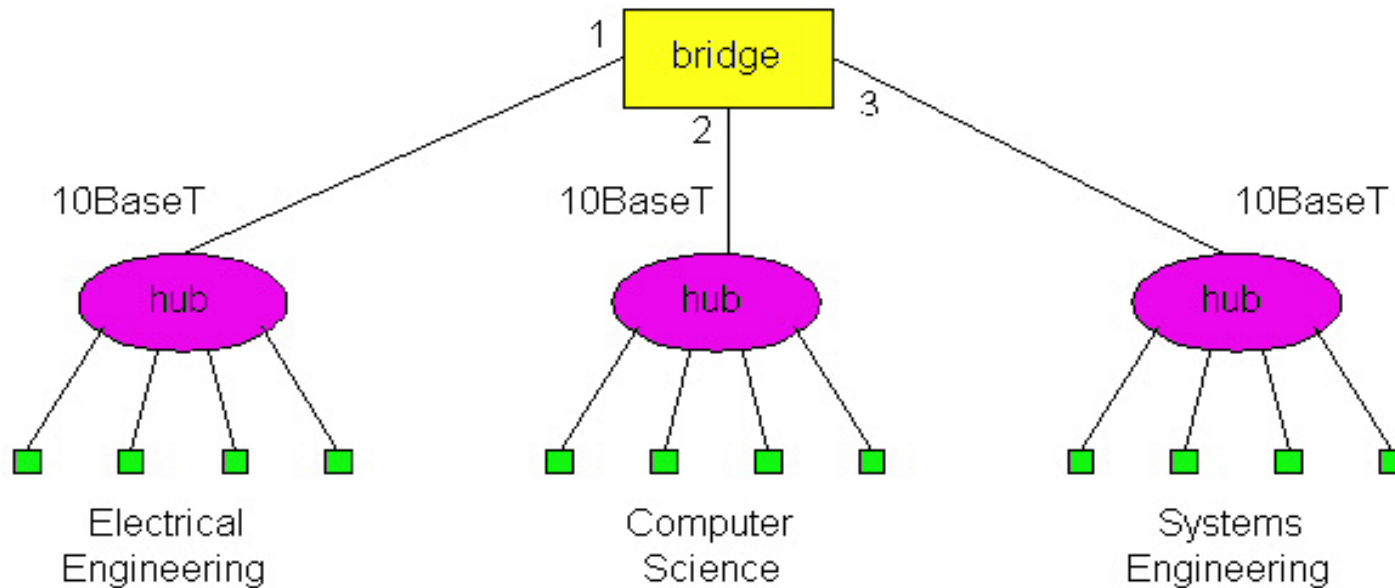


Recommended !

# Bridges Spanning Tree

□ for increased reliability, desirable to have redundant, alternative paths from source to dest

□ with multiple paths, cycles result - bridges may multiply and forward frame forever

□ solution: organize bridges in a spanning tree by disabling subset of interfaces



Disabled
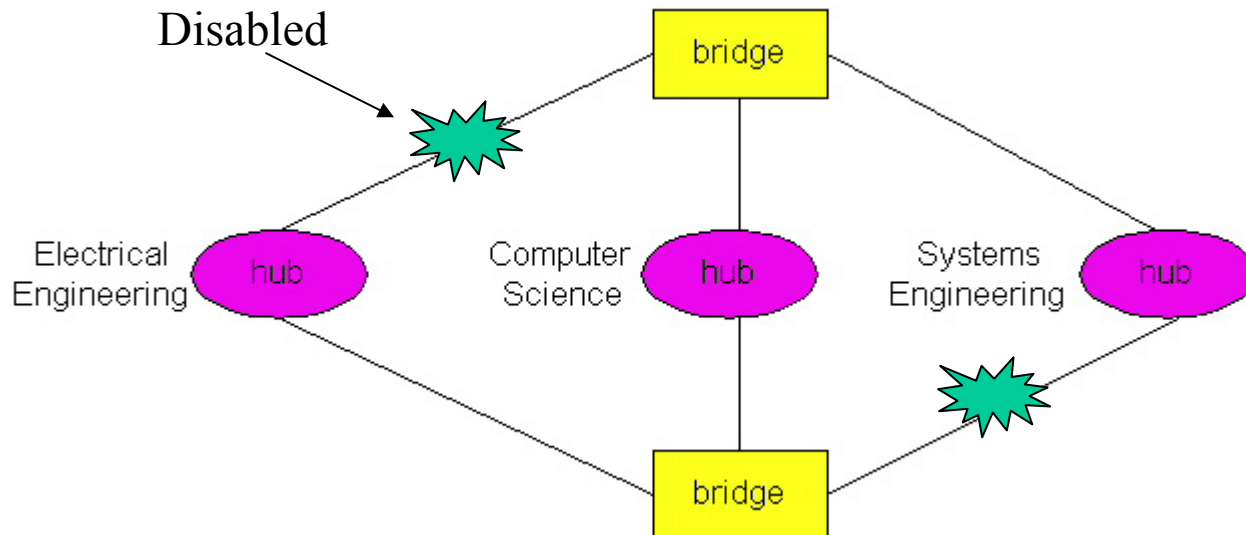
# Some bridge features

❒ Isolates collision domains resulting in higher total max throughput

❒ limitless number of nodes and geographical coverage

❒ Can connect different Ethernet types

❒ Transparent ("plug-and-play"): no configuration necessary

# Bridges vs. Routers

□ both store-and-forward devices
  ○ routers: network layer devices (examine network layer headers)
  ○ bridges are link layer devices
□ routers maintain routing tables, implement routing algorithms
□ bridges maintain bridge tables, implement filtering, learning and spanning tree algorithms



Host    Bridge    Router    Host
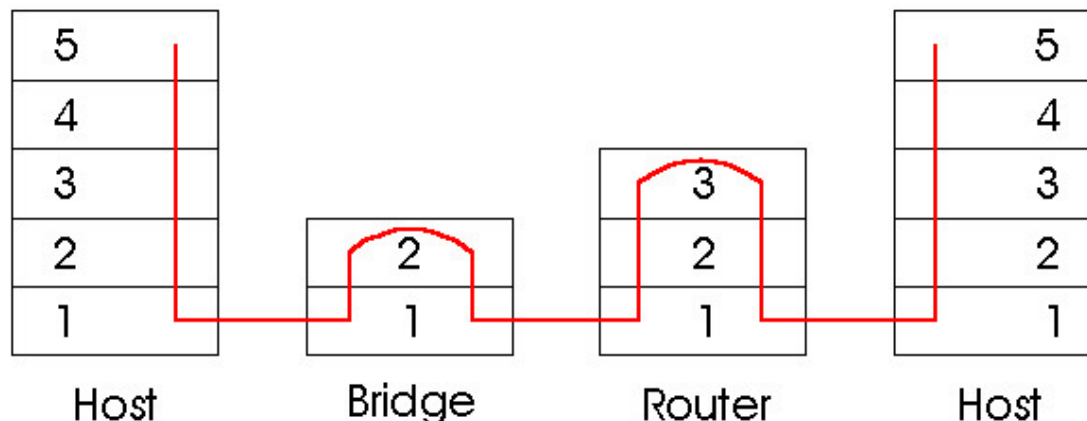
# Routers vs. Bridges

Bridges + and -

+ Bridge operation is simpler requiring less packet processing

+ Bridge tables are self learning

- All traffic confined to spanning tree, even when alternative bandwidth is available

- Bridges do not offer protection from broadcast storms

# Routers vs. Bridges

## Routers + and -

+ arbitrary topologies can be supported, cycling is limited by TTL counters (and good routing protocols)

+ provide protection against broadcast storms

- require IP address configuration (not plug and play)

- require higher packet processing

□ bridges do well in small (few hundred hosts) while routers used in large networks (thousands of hosts)

# Ethernet Switches

- Essentially a multi-interface bridge
- layer 2 (frame) forwarding, filtering using LAN addresses
- Switching: A-to-A' and B-to-B' simultaneously, no collisions
- large number of interfaces
- often: individual hosts, star-connected into switch
  - Ethernet, but no collisions!
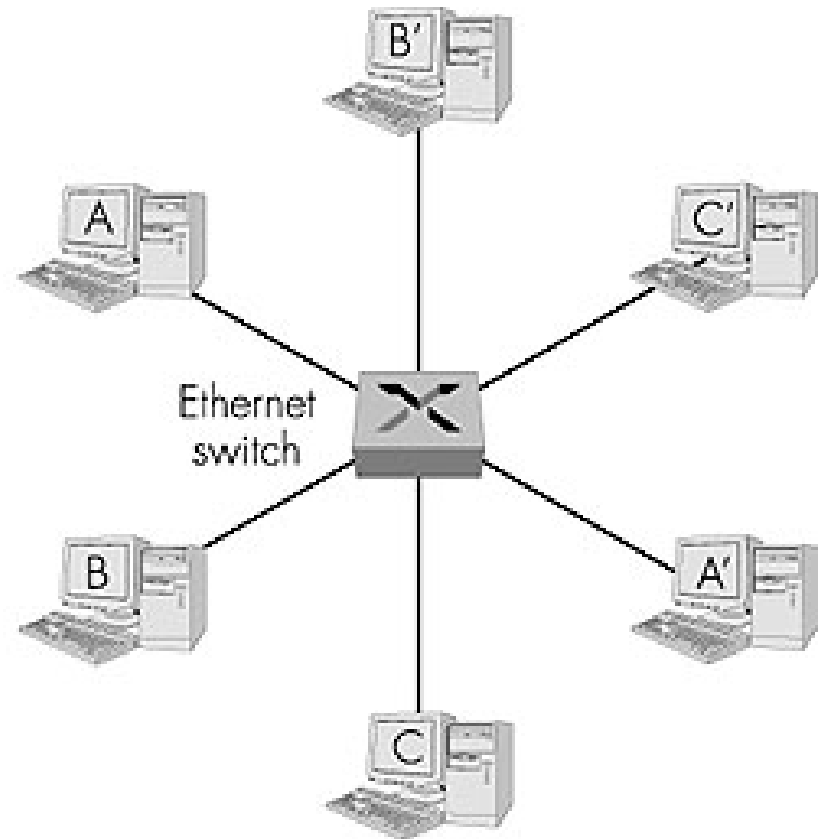


Ethernet switch

# Ethernet Switches

❒ cut-through switching: frame forwarded from input to output port without awaiting for assembly of entire frame

   ❍ slight reduction in latency

❒ combinations of shared/dedicated, 10/100/1000 Mbps interfaces

# Not an atypical LAN (IP network)
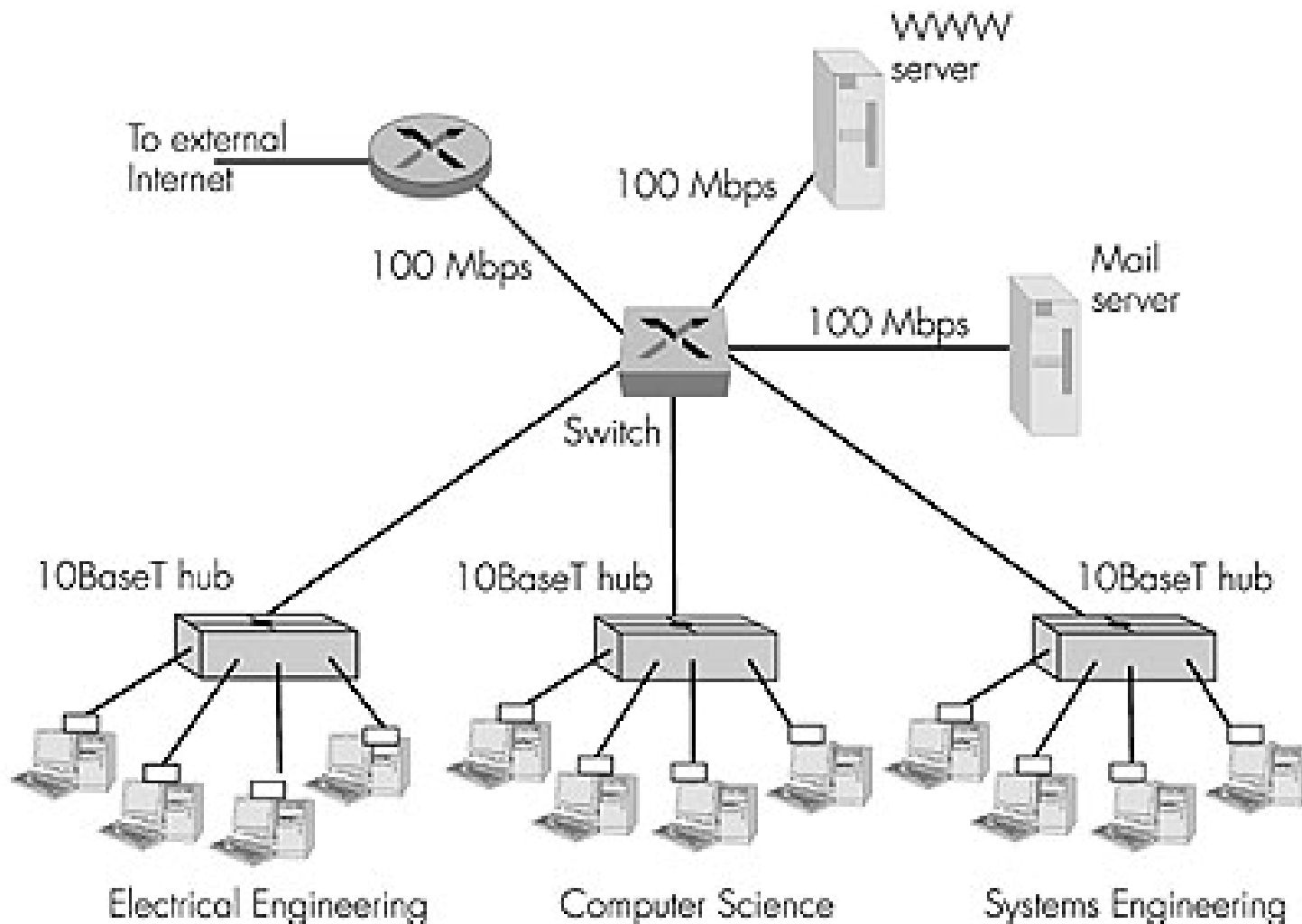
# Summary comparison

| | hubs | bridges | routers | switches |
|---|---|---|---|---|
| traffic isolation | no | yes | yes | yes |
| plug & play | yes | yes | no | yes |
| optimal routing | no | no | yes | no |
| cut through | yes | no | no | yes |

# Chapter 5 outline

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 LAN addresses and ARP
- 5.5 Ethernet

- 5.6 Hubs, bridges, and switches
- 5.7 Wireless links and LANs
- 5.8 PPP
- 5.9 ATM
- 5.10 Frame Relay

# IEEE 802.11 Wireless LAN

❒ **802.11b**
- ○ 2.4-5 GHz unlicensed radio spectrum
- ○ up to 11 Mbps
- ○ direct sequence spread spectrum (DSSS) in physical layer
  - • all hosts use same chipping code
- ○ widely deployed, using base stations

❒ **802.11a**
- ○ 5-6 GHz range
- ○ up to 54 Mbps

❒ **802.11g**
- ○ 2.4-5 GHz range
- ○ up to 54 Mbps

❒ All use CSMA/CA for multiple access

❒ All have base-station and ad-hoc network versions

# Base station approach

☐ Wireless host communicates with a base station
  ○ base station = access point (AP)
☐ Basic Service Set (BSS) (a.k.a. "cell") contains:
  ○ wireless hosts
  ○ access point (AP): base station
☐ BSSs combined to form distribution system (DS)
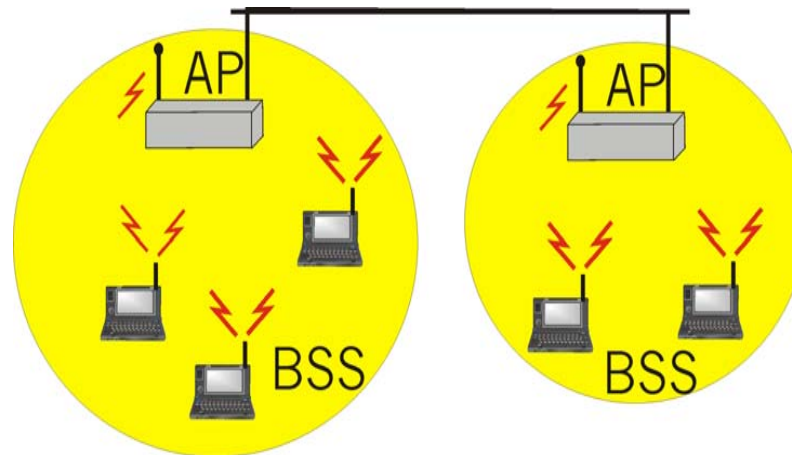
# Ad Hoc Network approach

- No AP (i.e., base station)
- wireless hosts communicate with each other
  - to get packet from wireless host A to B may need to route through wireless hosts X,Y,Z
- Applications:
  - "laptop" meeting in conference room, car
  - interconnection of "personal" devices
  - battlefield
- IETF MANET
  (Mobile Ad hoc Networks)
  working group
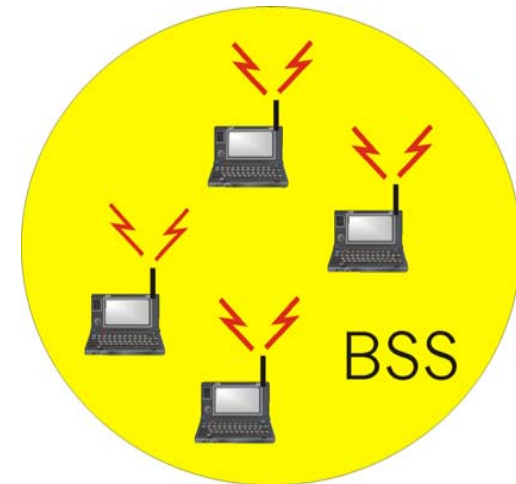
# IEEE 802.11: multiple access

- Collision if 2 or more nodes transmit at same time
- CSMA makes sense:
  - get all the bandwidth if you're the only one transmitting
  - shouldn't cause a collision if you sense another transmission
- Collision detection doesn't work: hidden terminal problem
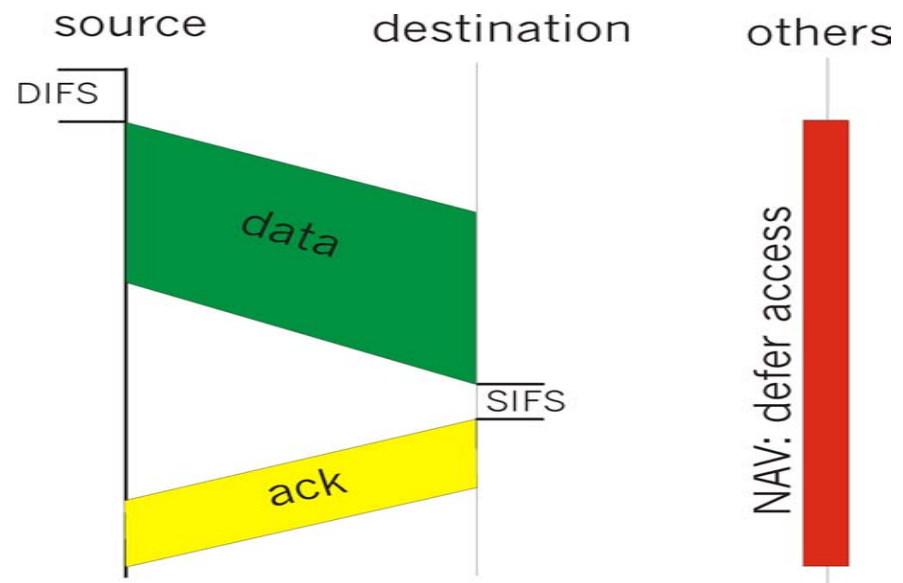


(a)          (b)
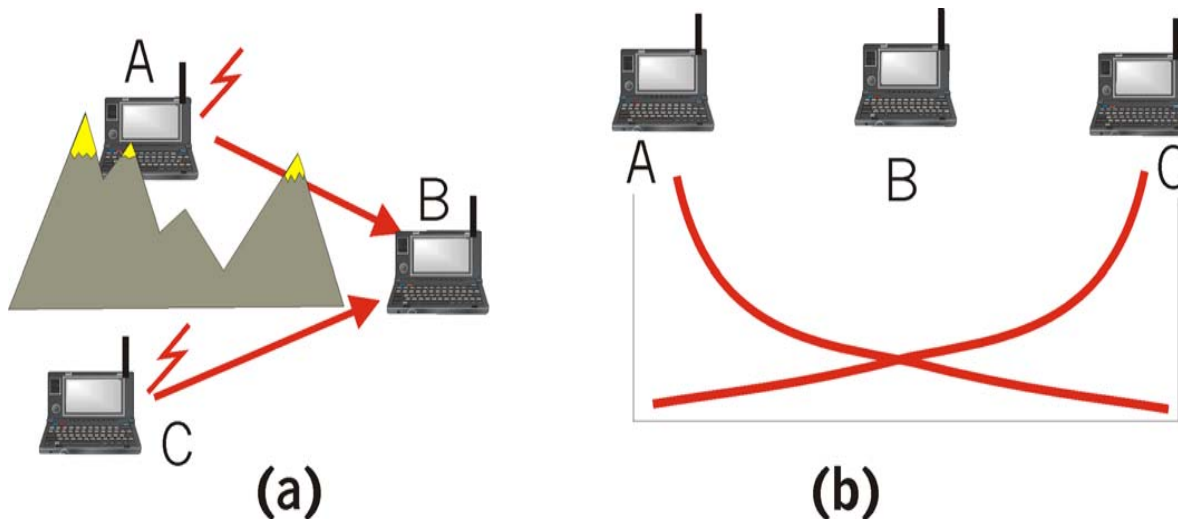
# IEEE 802.11 MAC Protocol: CSMA/CA

802.11 CSMA: sender

- if sense channel idle for **DISF** sec.

  then transmit entire frame (no collision detection)

- if sense channel busy then binary backoff

802.11 CSMA receiver

- if received OK

  return ACK after **SIFS**

  (ACK is needed due to hidden terminal problem)
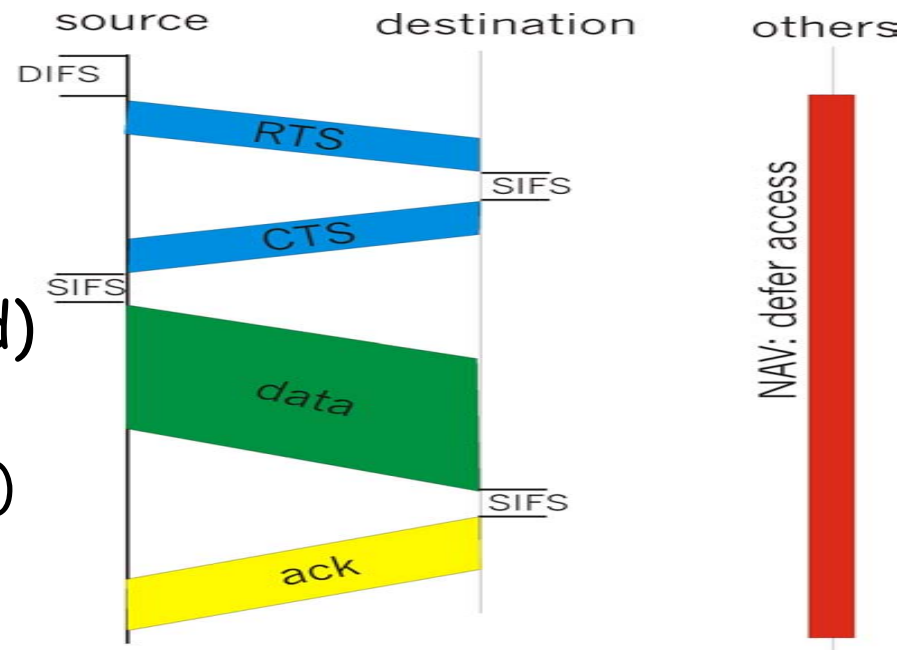
# Collision avoidance mechanisms

□ Problem:
  ○ two nodes, hidden from each other, transmit complete frames to base station
  ○ wasted bandwidth for long duration !

□ Solution:
  ○ small reservation packets
  ○ nodes track reservation interval with internal "network allocation vector" (NAV)
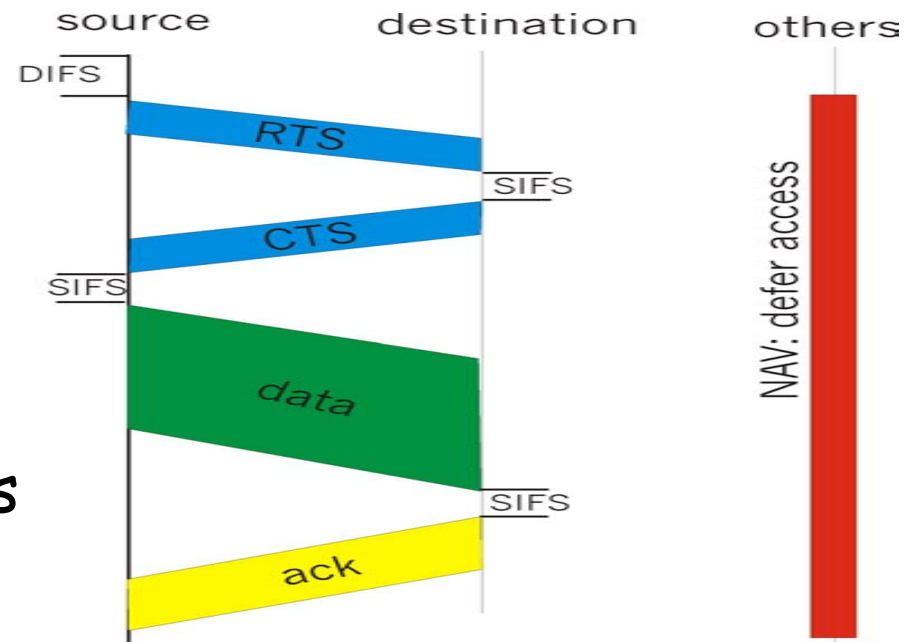
# Collision Avoidance: RTS-CTS exchange

❒ sender transmits short RTS (request to send) packet: indicates duration of transmission

❒ receiver replies with short CTS (clear to send) packet

   ○ notifying (possibly hidden) nodes

❒ hidden nodes will not transmit for specified duration: NAV



source    destination    others

DIFS

RTS

SIFS

CTS

SIFS

data

SIFS

ack

NAV: defer access

# Collision Avoidance: RTS-CTS exchange

- RTS and CTS short:
  - collisions less likely, of shorter duration
  - end result similar to collision detection
- IEEE 802.11 allows:
  - CSMA
  - CSMA/CA: reservations
  - polling from AP

source            destination            others

DIFS

RTS

SIFS

CTS

SIFS

data

SIFS

ack

NAV: defer access

# A word about Bluetooth

- Low-power, small radius, wireless networking technology
  - 10-100 meters
- omnidirectional
  - not line-of-sight infrared
- Interconnects gadgets
- 2.4-2.5 GHz unlicensed radio band
- up to 721 kbps

- Interference from wireless LANs, digital cordless phones, microwave ovens:
  - frequency hopping helps
- MAC protocol supports:
  - error correction
  - ARQ
- Each node has a 12-bit address

# Chapter 5 outline

# 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!

# PPP Design Requirements [RFC 1557]

❑ packet framing: encapsulation of network-layer datagram in data link frame
  ○ carry network layer data of any network layer protocol (not just IP) *at same time*
  ○ ability to demultiplex upwards
❑ bit transparency: must carry any bit pattern in the data field
❑ error detection (no correction)
❑ connection liveness: detect, signal link failure to network layer
❑ network layer address negotiation: endpoint can learn/configure each other's network address

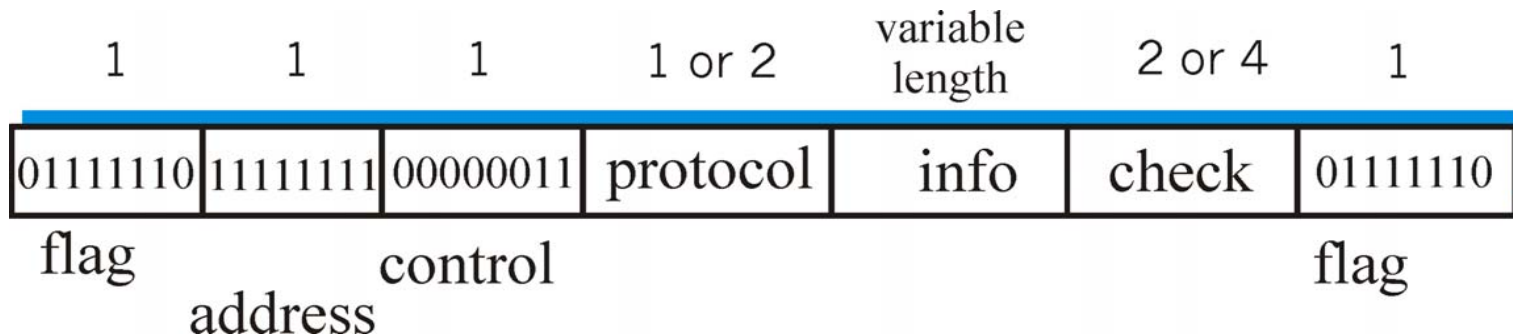# PPP non-requirements

☐ no error correction/recovery

☐ no flow control

☐ out of order delivery OK

☐ no need to support multipoint links (e.g., polling)

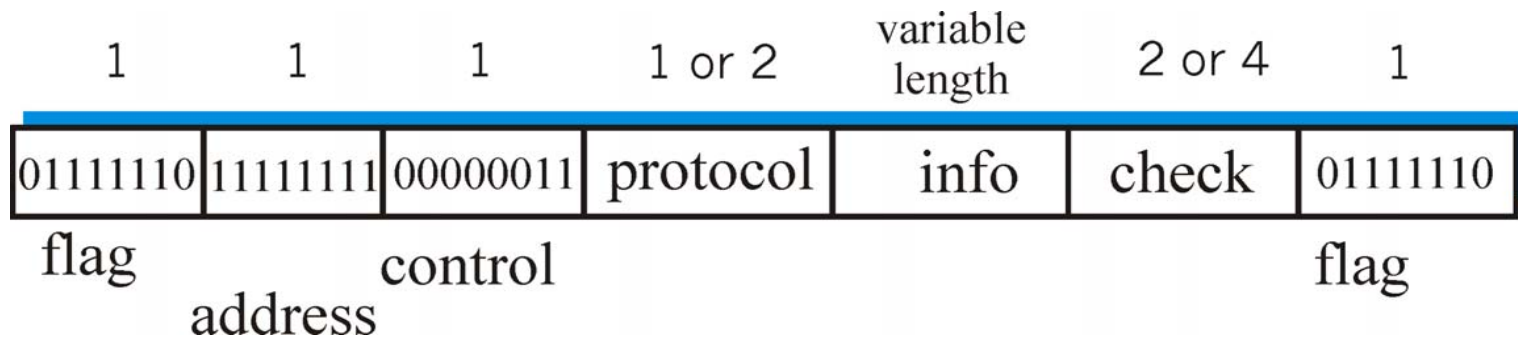Error recovery, flow control, data re-ordering
all relegated to higher layers!

# PPP Data Frame

❑ Flag: delimiter (framing)
❑ Address:  does nothing (only one option)
❑ Control: does nothing; in the future possible multiple control fields
❑ Protocol: upper layer protocol to which frame delivered (eg, PPP-LCP, IP, IPCP, etc)

| 1 | 1 | 1 | 1 or 2 | variable length | 2 or 4 | 1 |
|---|---|---|---|---|---|---|
| 01111110 | 11111111 | 00000011 | protocol | info | check | 01111110 |
| flag | address | control | | | | flag |

# PPP Data Frame

☐ info: upper layer data being carried
☐ check:  cyclic redundancy check for error detection

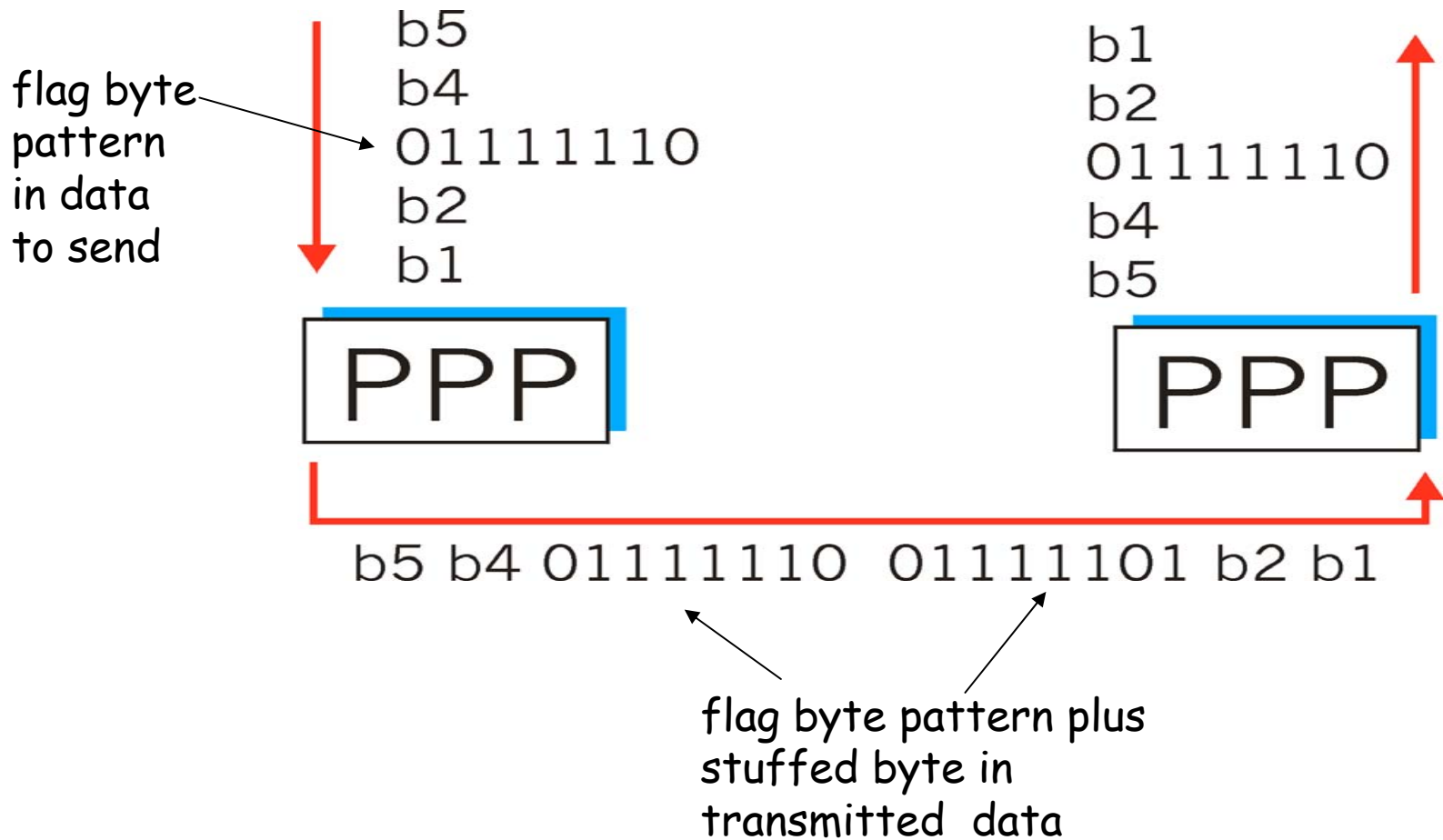| 1 | 1 | 1 | 1 or 2 | variable length | 2 or 4 | 1 |
|---|---|---|--------|-----------------|--------|---|
| 01111110 | 11111111 | 00000011 | protocol | info | check | 01111110 |
| flag | address | control | | | | flag |

# Byte Stuffing

□ "data transparency" requirement: data field must be allowed to include flag pattern <01111110>

○ Q: is received <01111110> data or flag?

□ Sender: adds ("stuffs") extra < 01111110> byte after each < 01111110> *data*  byte

□ Receiver:

○ two 01111110 bytes in a row: discard first byte, continue data reception

○ single 01111110: flag byte

# Byte Stuffing

flag byte
pattern
in data
to send

b5
b4
01111110
b2
b1

b1
b2
01111110
b4
b5

PPP

PPP

b5 b4 01111110 01111101 b2 b1

flag byte pattern plus
stuffed byte in
transmitted data

# PPP Data Control Protocol

Before exchanging network-layer data, data link peers must

□ configure PPP link (max. frame length, authentication)

□ learn/configure network layer information

   ○ for IP: carry IP Control Protocol (IPCP) msgs (protocol field: 8021) to configure/learn IP address
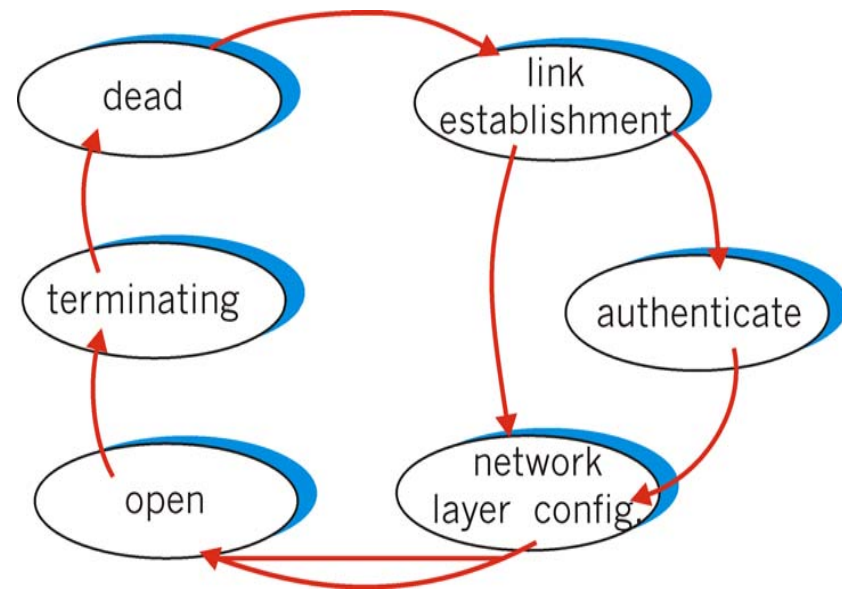
# Chapter 5 outline

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 LAN addresses and ARP
- 5.5 Ethernet

- 5.6 Hubs, bridges, and switches
- 5.7 Wireless links and LANs
- 5.8 PPP
- 5.9 ATM
- 5.10 Frame Relay

# Asynchronous Transfer Mode: ATM

□ **1990's/00 standard for high-speed** (155Mbps to 622 Mbps and higher) *Broadband Integrated Service Digital Network* architecture

□ Goal: *integrated, end-end transport of carry voice, video, data*

  ○ meeting timing/QoS requirements of voice, video (versus Internet best-effort model)

  ○ "next generation" telephony: technical roots in telephone world

  ○ packet-switching (fixed length packets, called "cells") using virtual circuits

# ATM architecture
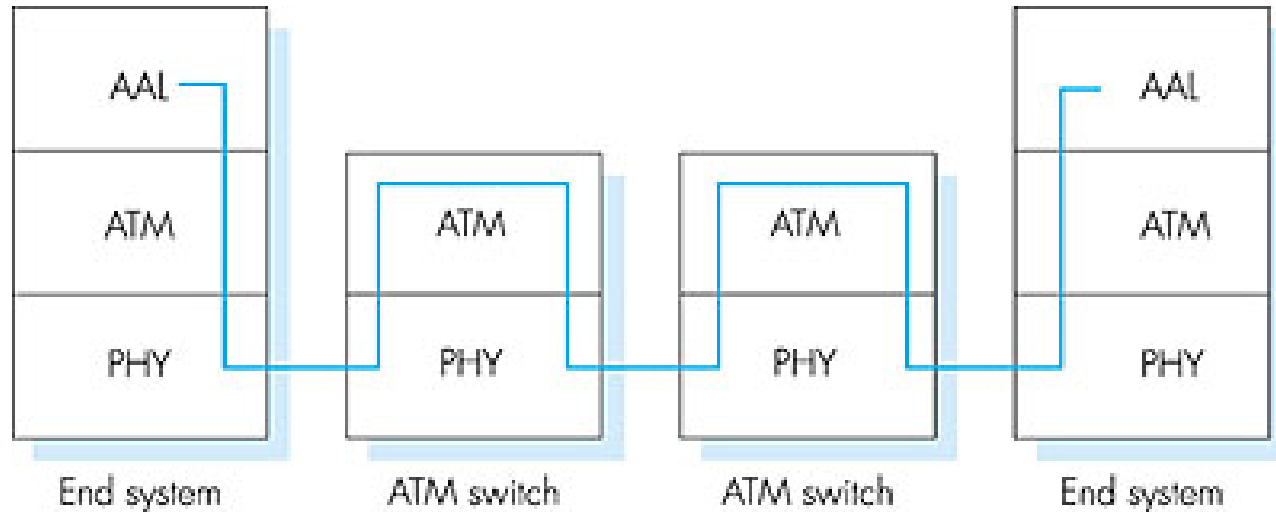


□ **adaptation layer:** only at edge of ATM network
  ❍ data segmentation/reassembly
  ❍ roughly analogous to Internet transport layer
□ **ATM layer:** "network" layer
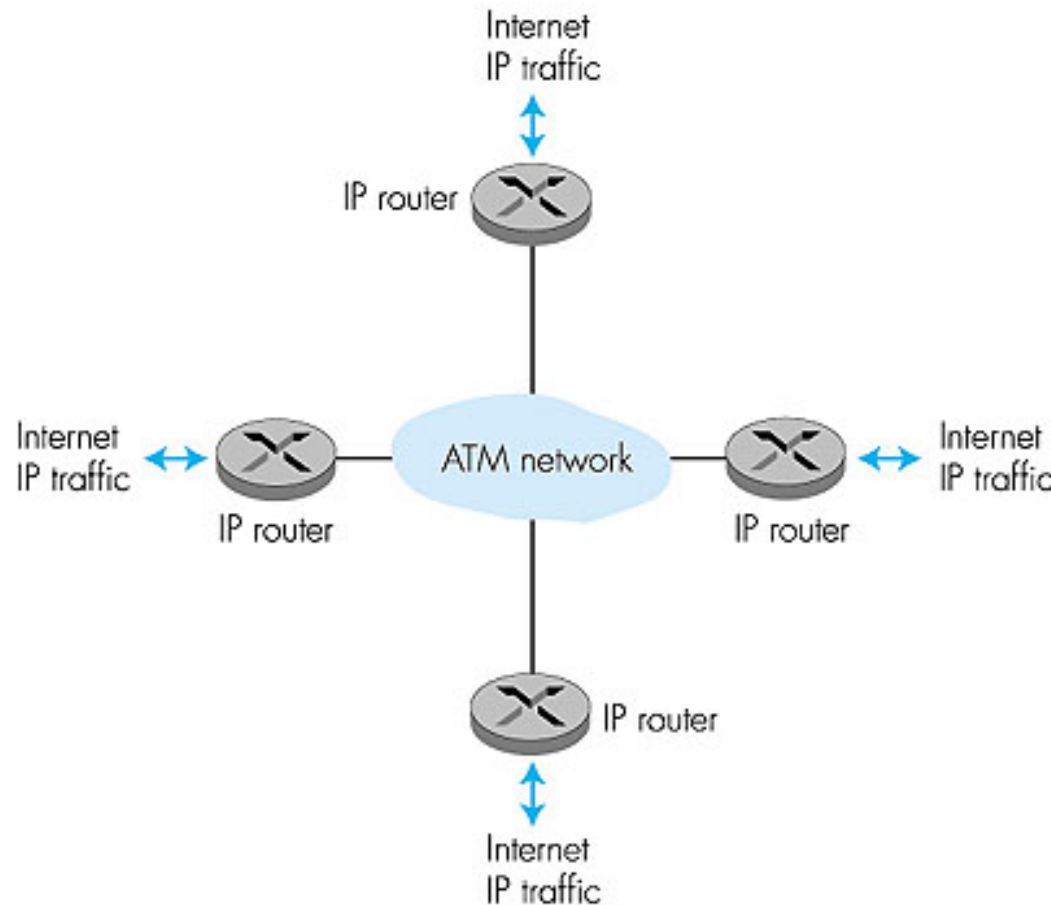  ❍ cell switching, routing
□ **physical layer**

# ATM: network or link layer?

Vision: end-to-end transport: "ATM from desktop to desktop"

  ❍ ATM *is* a network technology

Reality: used to connect IP backbone routers

  ❍ "IP over ATM"

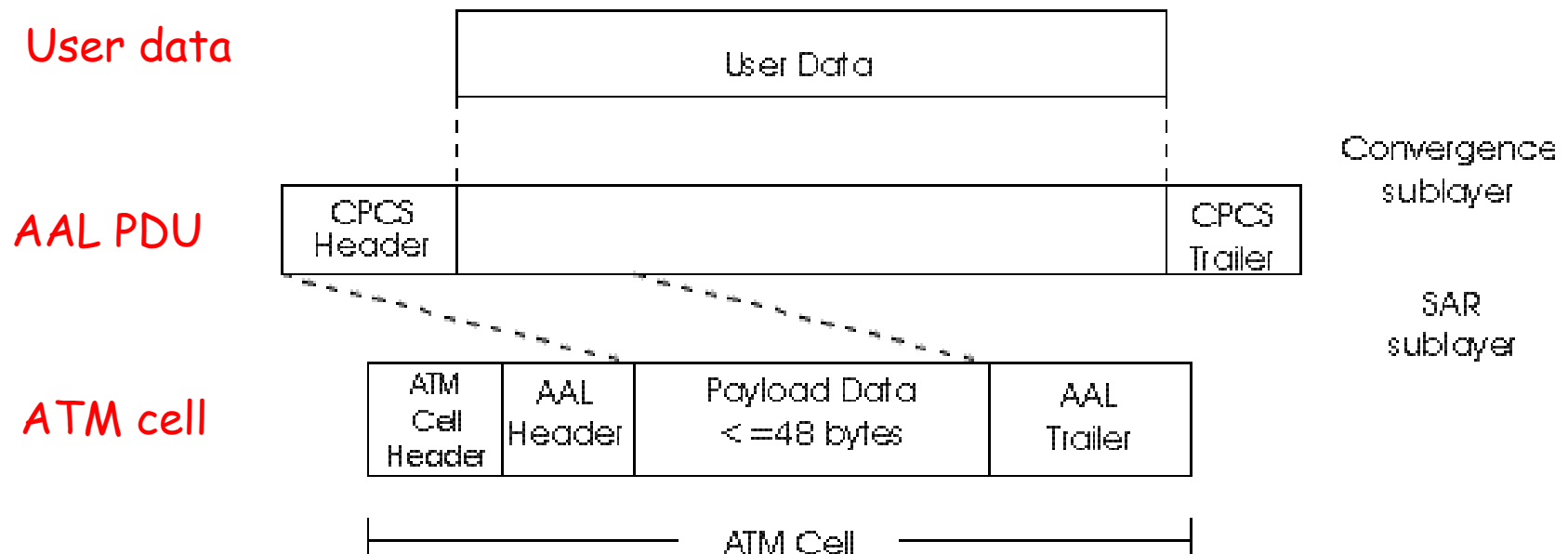  ❍ ATM as switched link layer, connecting IP routers

# ATM Adaptation Layer (AAL)

□ ATM **Adaptation Layer** (AAL): "adapts" upper layers (IP or native ATM applications)  to ATM layer below

□ AAL present **only in end systems**, not in switches

□ AAL layer segment (header/trailer fields, data) fragmented across multiple ATM cells

  ○ analogy: TCP segment in many IP packets
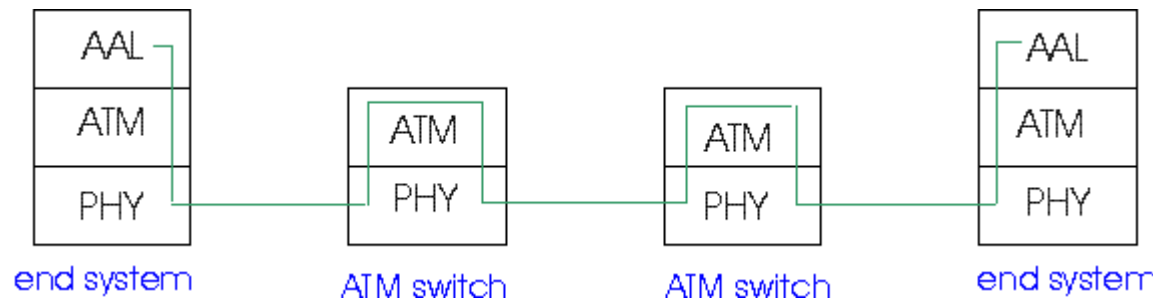
# ATM Adaptation Layer (AAL) [more]

Different versions of AAL layers, depending on ATM service class:

❒ AAL1: for CBR (Constant Bit Rate) services, e.g. circuit emulation
❒ AAL2: for VBR (Variable Bit Rate) services, e.g., MPEG video
❒ AAL5: for data (e.g., IP datagrams)

User data

| User Data |

AAL PDU

| CPCS Header | | CPCS Trailer |

Convergence sublayer

SAR sublayer

ATM cell

| ATM Cell Header | AAL Header | Payload Data <=48 bytes | AAL Trailer |

ATM Cell

# AAL5 - Simple And Efficient AL (SEAL)

☐ **AAL5**: **low overhead** AAL used to carry IP datagrams
  - ○ 4 byte cyclic redundancy check
  - ○ PAD ensures payload multiple of 48bytes
  - ○ large AAL5 data unit to be fragmented into 48-byte ATM cells

| CPCS-PDU payload | PAD | Length | CRC |
|---|---|---|---|
| 0-65535 | 0-47 | 2 | 4 |
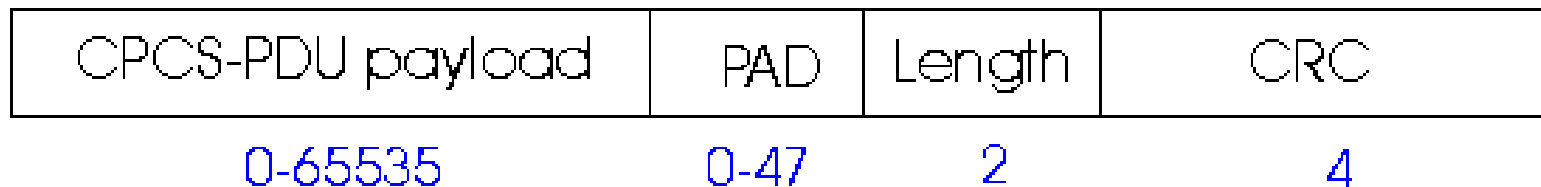
# ATM Layer

Service: transport cells across ATM network

❒ analogous to IP network layer

❒ very different services than IP network layer

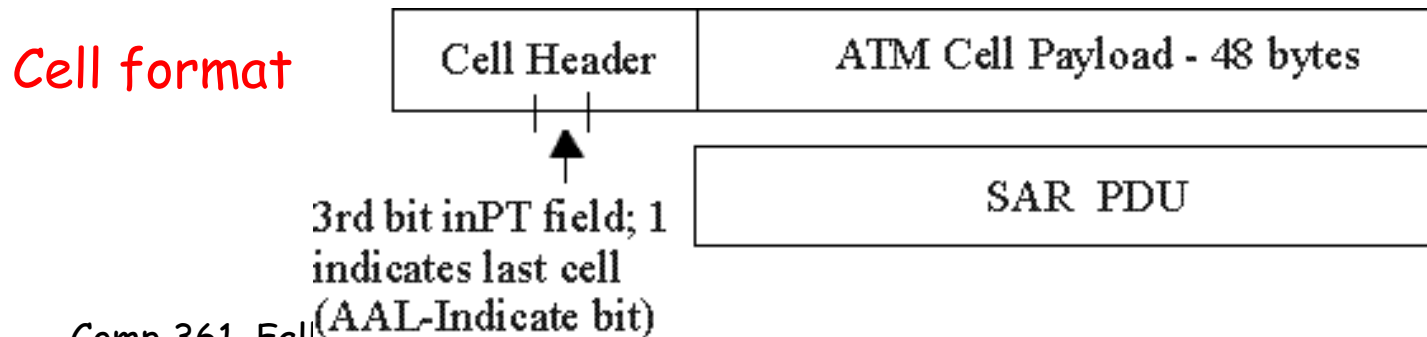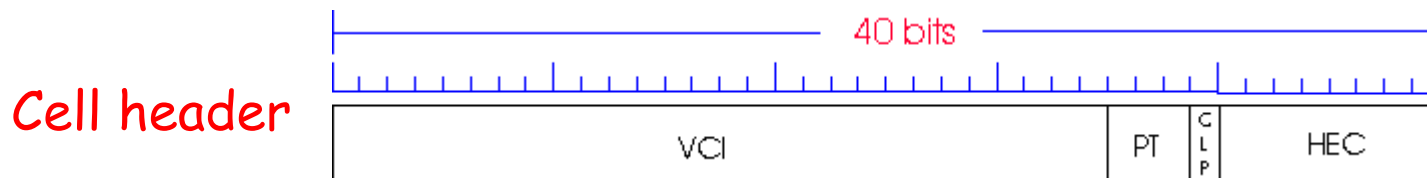| Network Architecture | Service Model | Guarantees ? | | | | Congestion feedback |
|---|---|---|---|---|---|---|
| | | Bandwidth | Loss | Order | Timing | |
| Internet | best effort | none | no | no | no | no (inferred via loss) |
| ATM | CBR | constant rate | yes | yes | yes | no congestion |
| ATM | VBR | guaranteed rate | yes | yes | yes | no congestion |
| ATM | ABR | guaranteed minimum | no | yes | no | yes |
| ATM | UBR | none | no | yes | no | no |

# ATM Layer: Virtual Circuits

□ **VC transport:** cells carried on VC from source to dest
  ❍ call setup, teardown for each call *before* data can flow
  ❍ each packet carries VC identifier (not destination ID)
  ❍ *every* switch on source-dest path maintain "state" for each passing connection
  ❍ link,switch resources (bandwidth, buffers) may be *allocated* to VC: to get circuit-like perf.

□ **Permanent VCs (PVCs)**

  ❍ long lasting connections
  ❍ typically: "permanent" route between to IP routers

□ **Switched VCs (SVC):**

  ❍ dynamically set up on per-call basis

# ATM VCs

- Advantages of ATM VC approach:
  - QoS performance guarantee for connection mapped to VC (bandwidth, delay, delay jitter)
- Drawbacks of ATM VC approach:
  - Inefficient support of datagram traffic
  - one PVC between each source/dest pair) does not scale (N*2 connections needed)
  - SVC introduces call setup latency, processing overhead for short lived connections

# ATM Layer: ATM cell

□ 5-byte ATM cell header

□ 48-byte payload
  ○ Why?: small payload -> short cell-creation delay for digitized voice
  ○ halfway between 32 and 64 (compromise!)

Cell header

40 bits

| VCI | PT | C L P | HEC |

Cell format

| Cell Header | ATM Cell Payload - 48 bytes |

SAR PDU

3rd bit inPT field; 1 indicates last cell (AAL-Indicate bit)
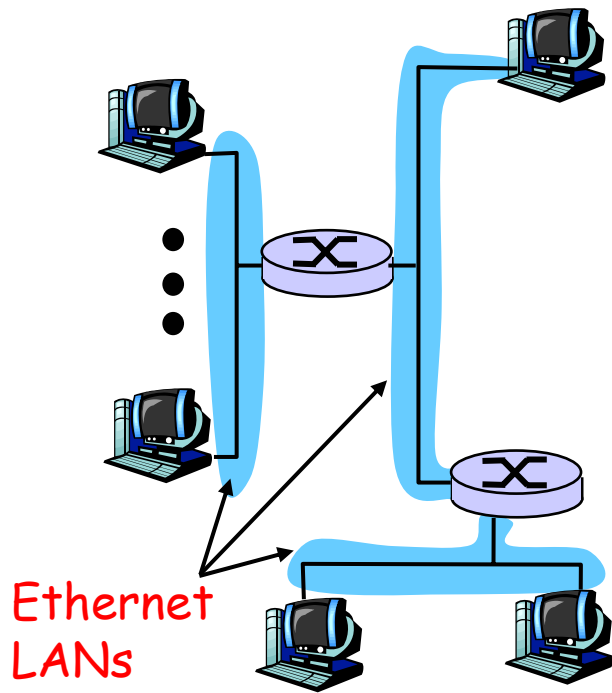
# ATM cell header

- **VCI:** virtual channel ID
  - will *change* from link to link thru net
- **PT:** Payload type (e.g. RM cell versus data cell)
- **CLP:** Cell Loss Priority bit
  - CLP = 1 implies low priority cell, can be discarded if congestion
- **HEC:** Header Error Checksum
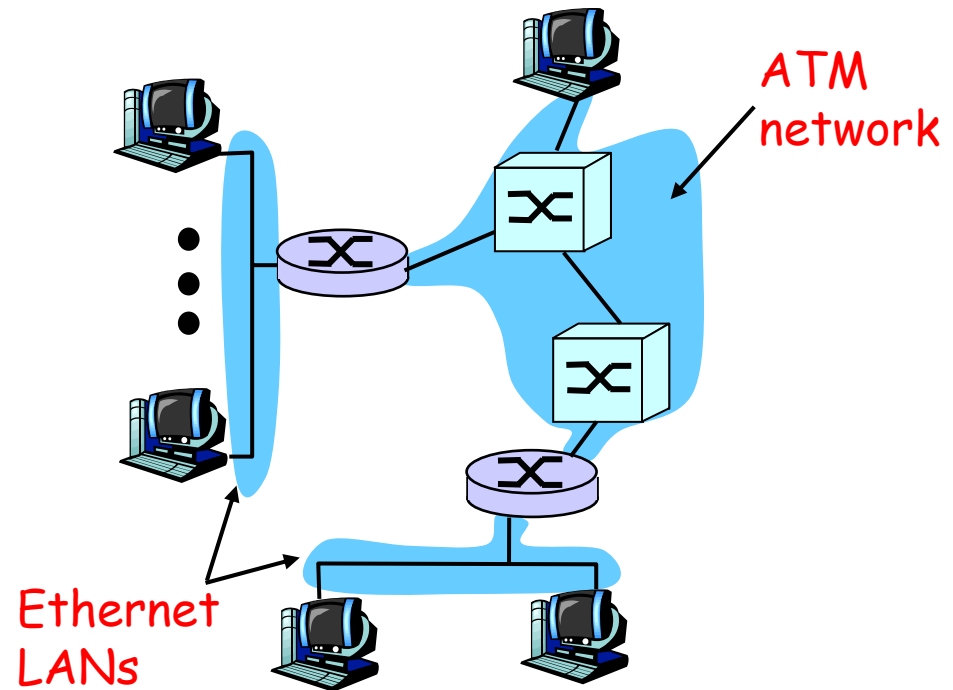  - cyclic redundancy check
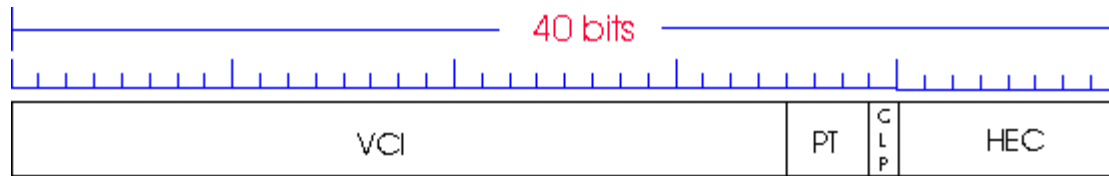
# IP-Over-ATM

## Classic IP only

□ 3 "networks" (e.g., LAN segments)
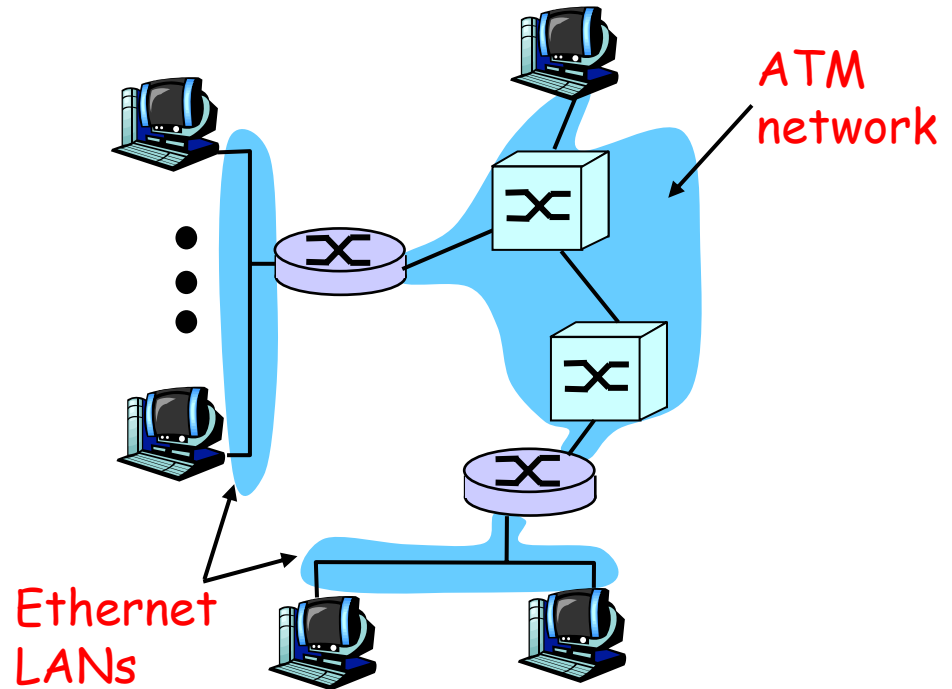
□ MAC (802.3) and IP addresses

## IP over ATM

□ replace "network" (e.g., LAN segment) with ATM network

□ ATM addresses, IP addresses

Ethernet LANs

ATM network

Ethernet LANs

# IP-Over-ATM

**Issues:**

☐ IP datagrams into ATM AAL5 PDUs

☐ from IP addresses to ATM addresses

   ○ just like IP addresses to 802.3 MAC addresses!



ATM network

Ethernet LANs

# Chapter 5 outline

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 LAN addresses and ARP
- 5.5 Ethernet

- 5.6 Hubs, bridges, and switches
- 5.7 Wireless links and LANs
- 5.8 PPP
- 5.9 ATM
- 5.10 Frame Relay

# Frame Relay

## Like ATM:

❒ wide area network technologies

❒ Virtual-circuit oriented

❒ origins in telephony world

❒ can be used to carry IP datagrams

   ❍ can thus be viewed as link layers by IP protocol

# Frame Relay

□ Designed in late '80s, widely deployed in the '90s

□ Frame relay service:

   ❍ no error control

   ❍ end-to-end congestion control
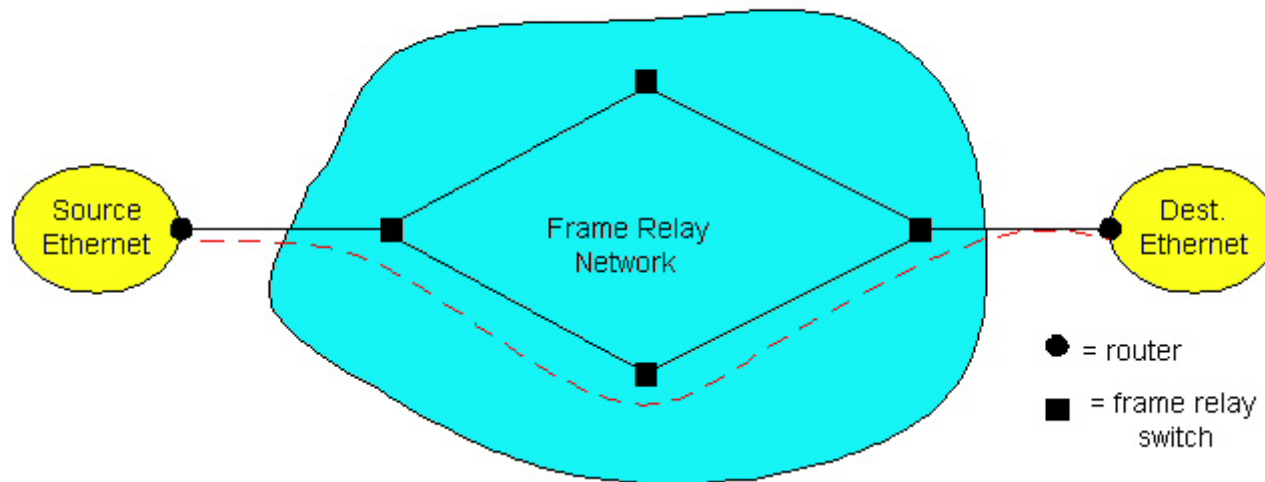
# Frame Relay (more)

- Designed to **interconnect** corporate customer LANs
  - typically permanent VC's: "**pipe**" carrying aggregate traffic between two routers
  - switched VC's: as in ATM
- corporate customer **leases** FR service from public Frame Relay network (e.g., Sprint, ATT)

layer 3 — User data field

layer 2 — Flag | Link layer | | CRC | Flag
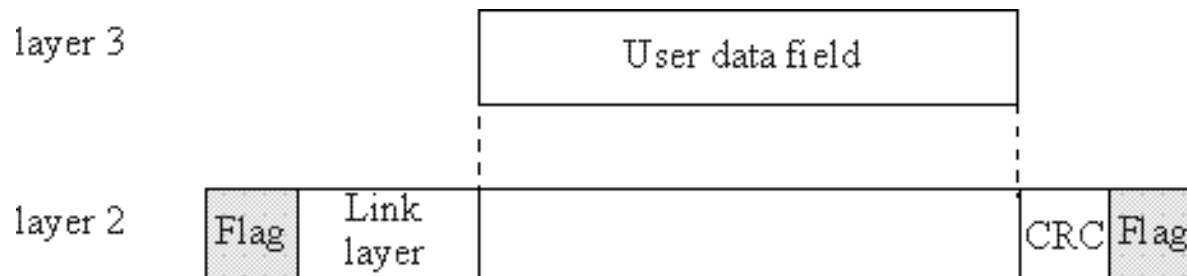
Frame addressing and routing, congestion notification

# Frame Relay (more)

| flags | address | data | CRC | flags |
|-------|---------|------|-----|-------|

☐ Flag bits, 01111110, delimit frame

☐ address:
  ❍ 10 bit VC ID field
  ❍ 3 congestion control bits
    • FECN: forward explicit congestion notification (frame experienced congestion on path)
    • BECN: congestion on reverse path
    • DE: discard eligibility

# Frame Relay -VC Rate Control

□ Committed Information Rate (CIR)
  ○ defined, "guaranteed" for each VC
  ○ negotiated at VC set up time
  ○ customer pays based on CIR

□ DE bit: Discard Eligibility bit
  ○ Edge FR switch measures traffic rate for each VC; marks DE bit
  ○ DE = 0: high priority, rate compliant frame; deliver at "all costs"
  ○ DE = 1: low priority, eligible for congestion discard

# Frame Relay - CIR & Frame Marking

- **Access Rate**: rate **R** of the access link between **source router** (customer) and **edge FR switch** (provider); 64Kbps < **R** < 1,544Kbps
- Typically, **many VCs** (one per destination router) multiplexed on the same access trunk; each VC has own **CIR**
- Edge FR switch **measures** traffic rate for each VC; it **marks** (i.e. DE = 1) frames which **exceed** CIR (these may be later dropped)
- Internet's more recent differentiated service uses similar ideas

# Chapter 5: Summary

□ **principles** behind data link layer services:
- ○ error detection, correction
- ○ sharing a broadcast channel: multiple access
- ○ link layer addressing, ARP

□ **link layer technologies:** Ethernet, hubs, bridges, switches,IEEE 802.11 LANs, PPP, ATM, Frame Relay

□ journey down the protocol stack now *OVER!*

□ for continuation, e.g., multicasting, multimedia, enroll in comp362!