CDA 4506
Design and Implementation of Data Communication Networks

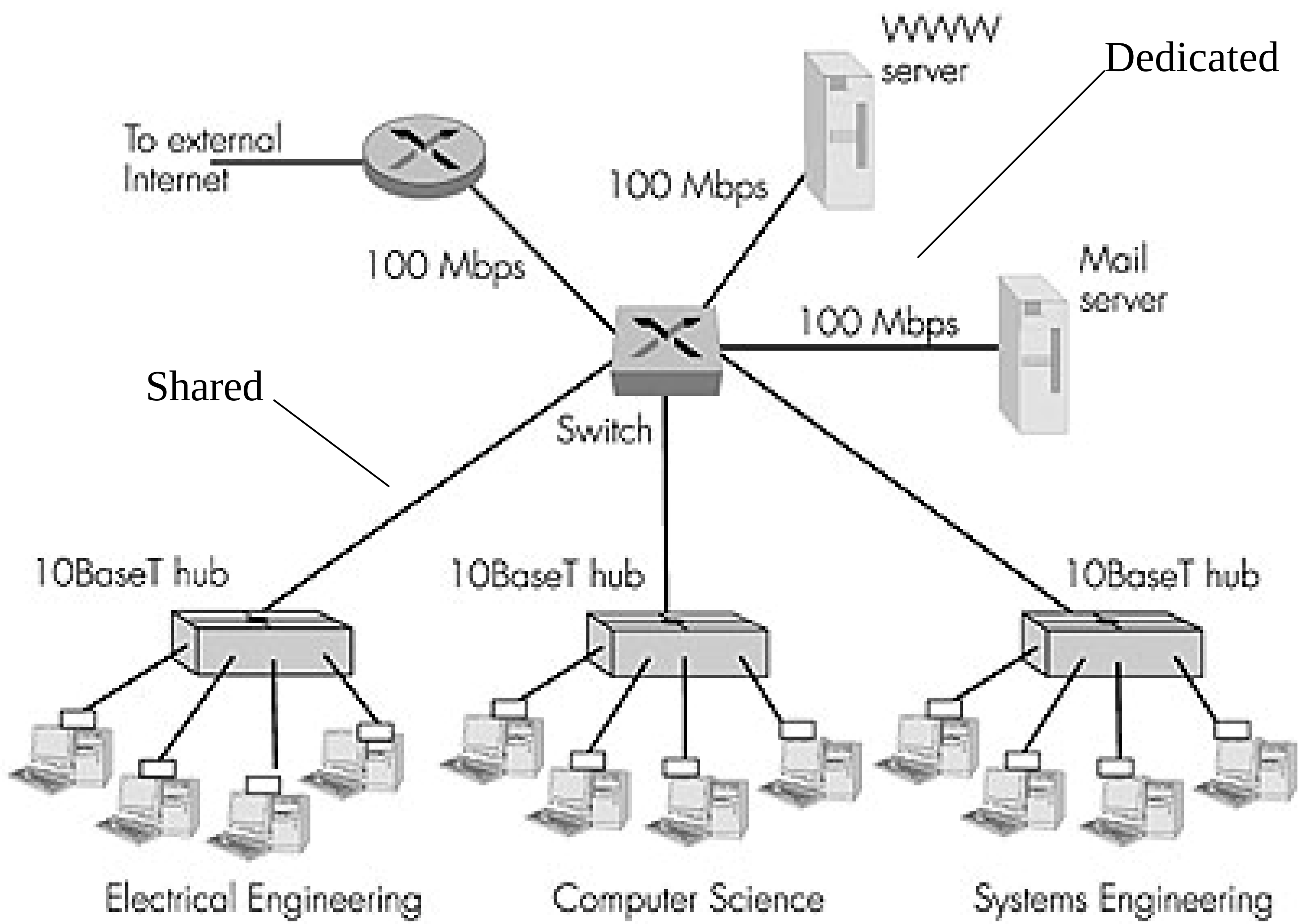
Lecture Set 2

Dr. R. Lent

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

Not an atypical LAN (IP network)



Chapter 5: The Data Link Layer

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 (Ch. 3)
- instantiation and implementation of various link layer technologies

Layered Architecture

Application	SMTP, FTP, HTTP, etc
Transport	TCP, UDP
Network	IP, ICMP
Link Physical	Ethernet, WiFi, PPP, ATM, Frame Relay, X.25

Datagrams

2-PDU is often called a **frame** **(**encapsulates datagram)

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

<u>transportation analogy</u>

- 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

1. 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!

2. Reliable delivery between adjacent nodes
   - 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)

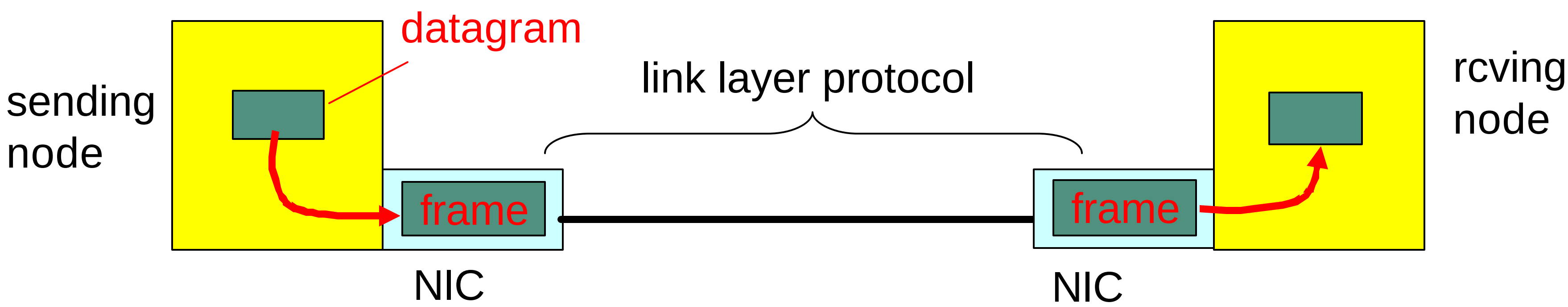
3. *Flow Control:*
   - pacing between adjacent sending and receiving nodes

4. *Error Detection*:
   - errors caused by signal attenuation, noise.
   - receiver detects presence of errors:
     - signals sender for retransmission or drops frame

5. Error Correction:
   - receiver identifies *and corrects* bit error(s) without resorting to retransmission

6. *Half-duplex or full-duplex*
   - with half duplex, nodes at both ends of link can transmit, but not at same time

Adaptors Communicating



- link layer implemented in a NIC (network interface card)
  - 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, reliability, flow control, etc
  - extracts datagram, passes to receiving node
- adapter is semi-autonomous
- link & physical layers

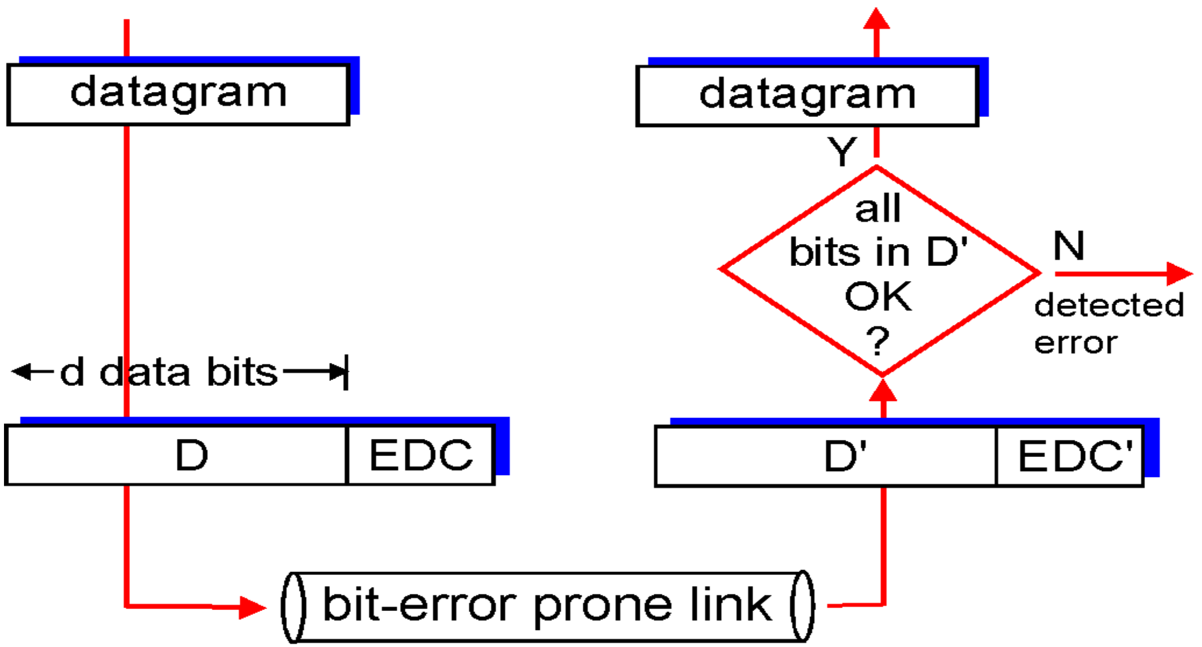
Chapter 5 outline

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3Multiple 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

Error Detection

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

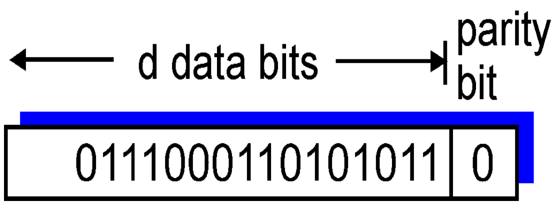
- Error detection not 100% reliable!
  - protocol may miss some errors, but rarely
  - larger EDC field yields better detection and correction



Parity Checking

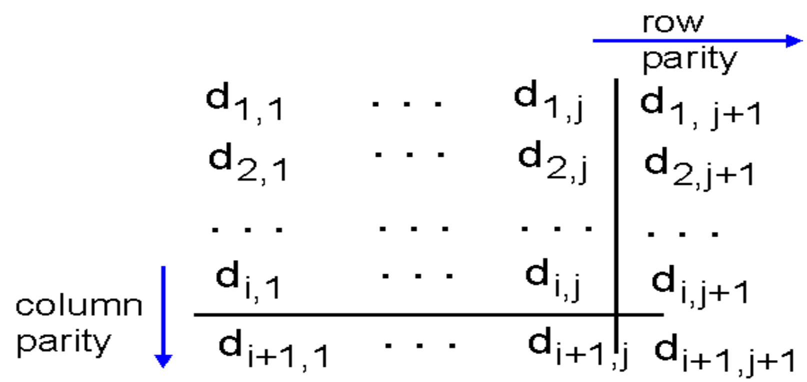
Single Bit Parity:

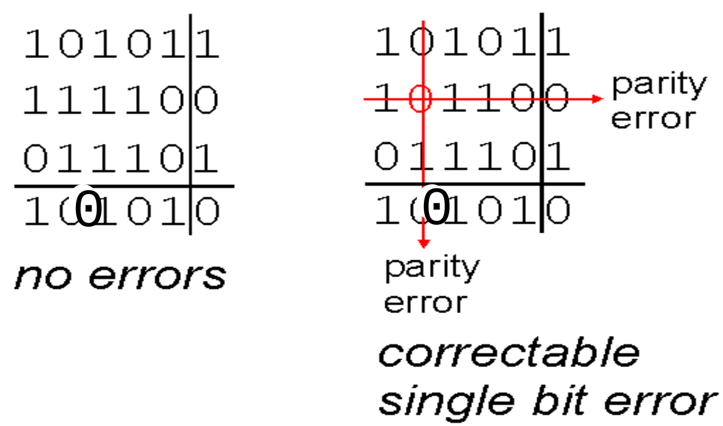
Detect single bit errors



Two Dimensional Bit Parity:

Detect *and correct* single bit errors





Internet checksum

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

Sender:

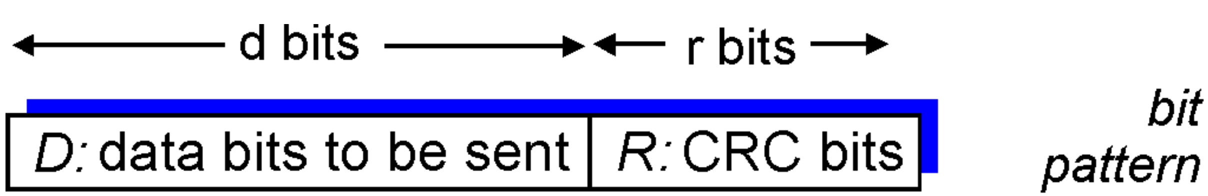
- treat segment contents as sequence of 16-bit integers
- checksum: addition (1's complement sum) of segment contents
- sender puts checksum value into UDP checksum field

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?* More later ….

Checksumming: Cyclic Redundancy Check

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



$D * 2^r$ XOR R *mathematical formula*

CRC Example

Want:

$D \cdot 2^r$ XOR R = nG

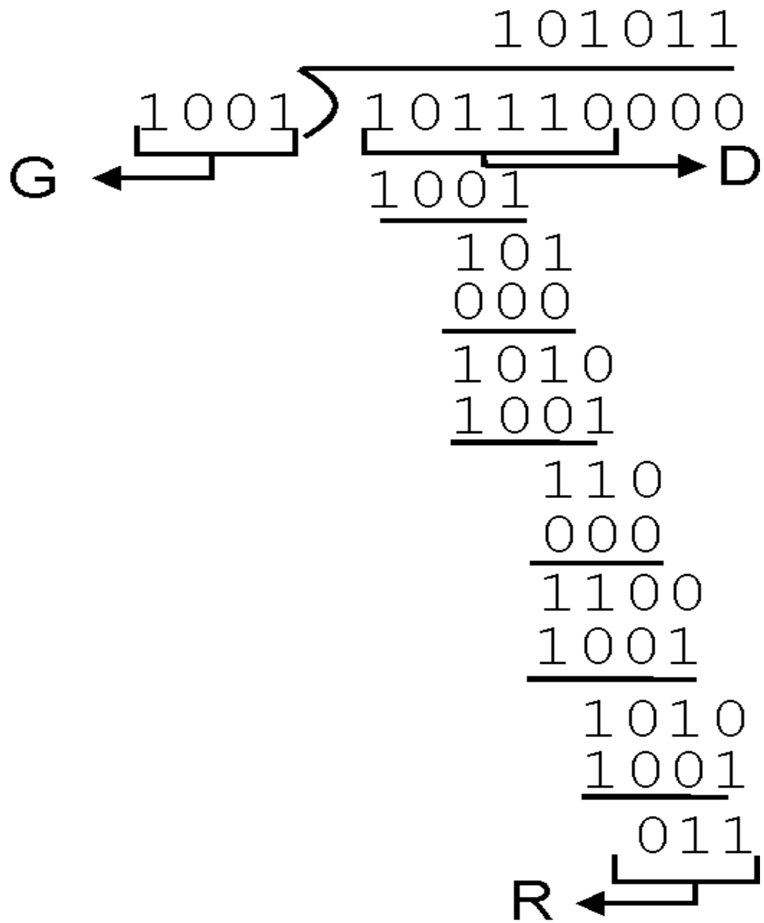
equivalently:

$D \cdot 2^r$ = nG XOR R

equivalently:

if we divide $D \cdot 2^r$ by G, want remainder R

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



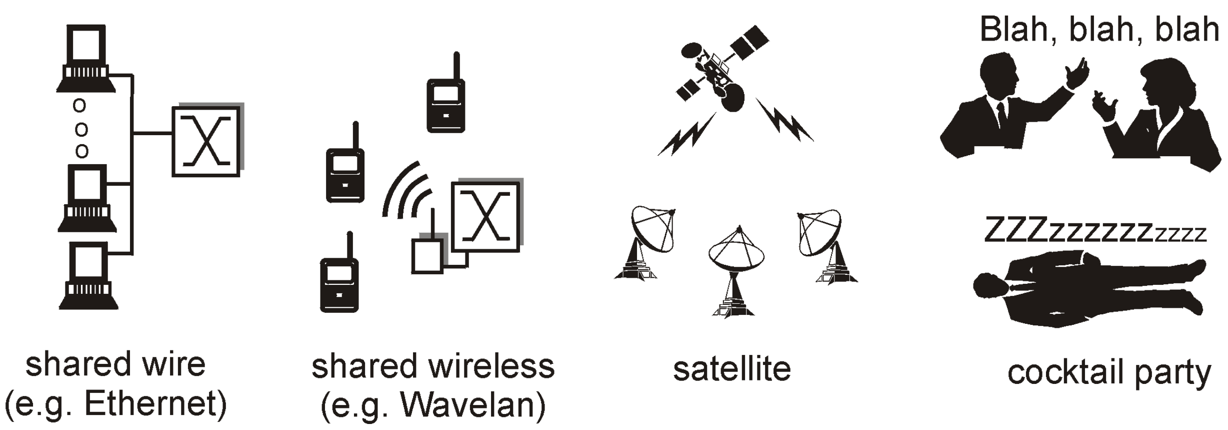
Chapter 5 outline

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3Multiple 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

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



Multiple Access protocols

- single shared broadcast 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 nodes share channel, i.e., determine when node can transmit
- communication about channel sharing must use channel itself!
- what to look for in multiple access protocols:

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:

1. Channel Partitioning
   - divide channel into smaller “pieces” (time slots, frequency, code)
   - allocate piece to node for exclusive use
2. Random Access
   - channel not divided, allow collisions
   - “recover” from collisions
3. “Taking turns”
   - tightly coordinate shared access to avoid collisions

Comparison of 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!

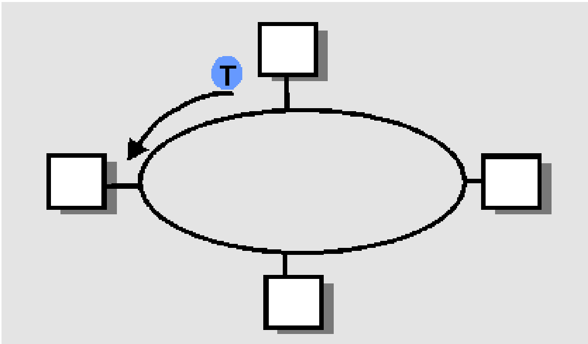
“Taking Turns” MAC protocols

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)



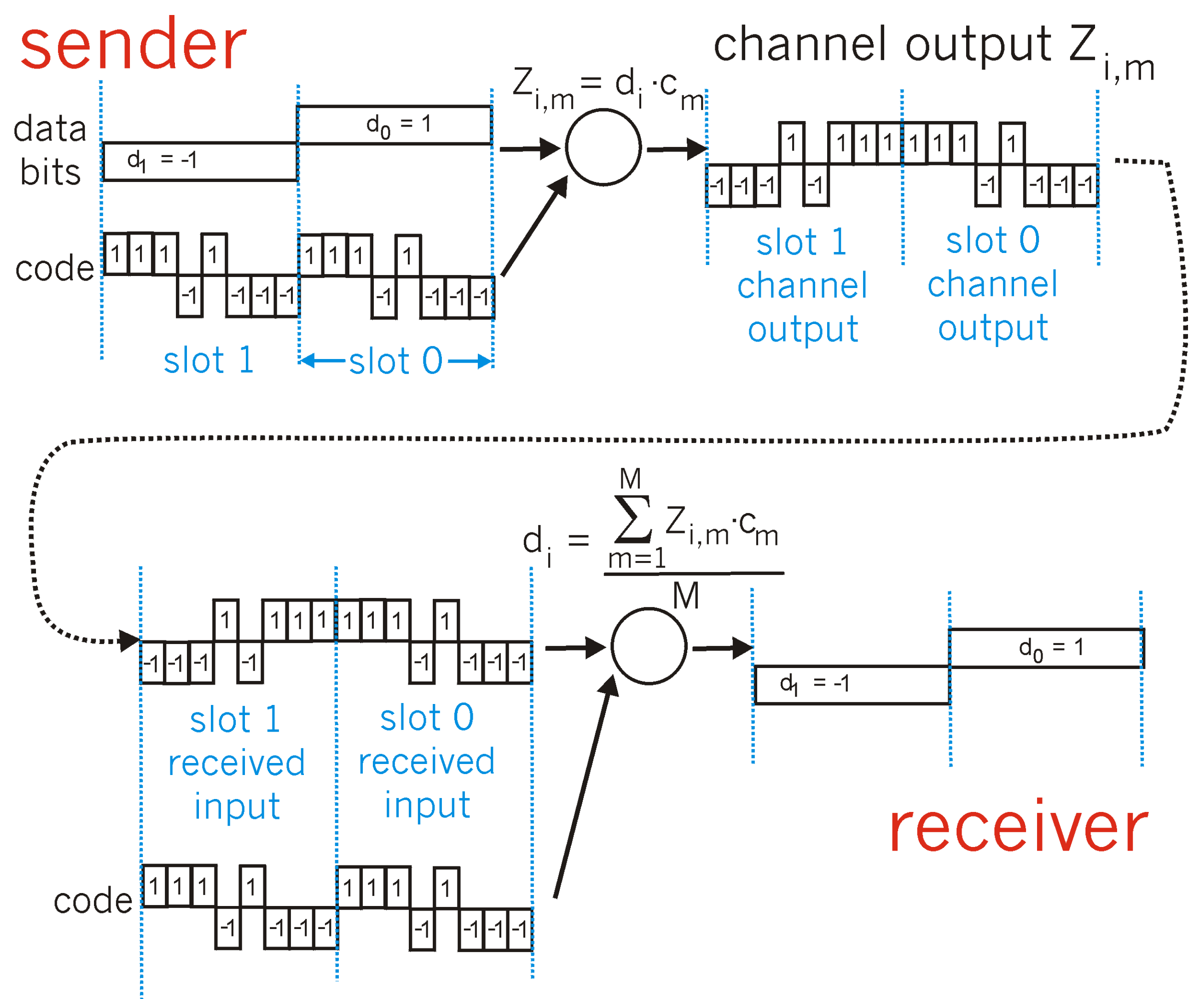
Channel Partitioning

We've studied TDMA and FDMA already

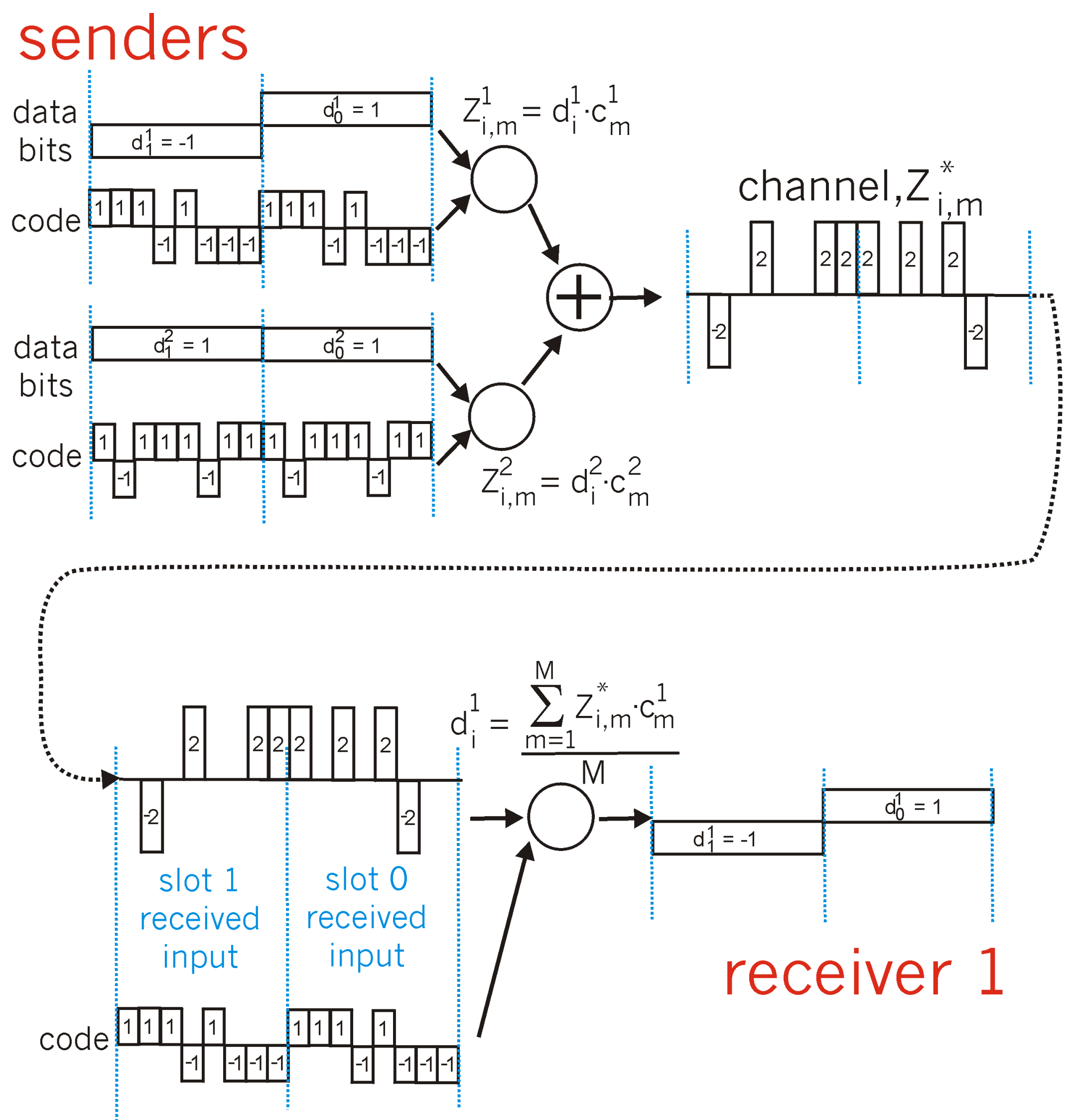
CDMA (Code Division Multiple Access) :

- 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 $\Rightarrow$ “collision”
- random access MAC protocol specifies:
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed retransmissions)
- Examples of random access MAC protocols:
  - slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA

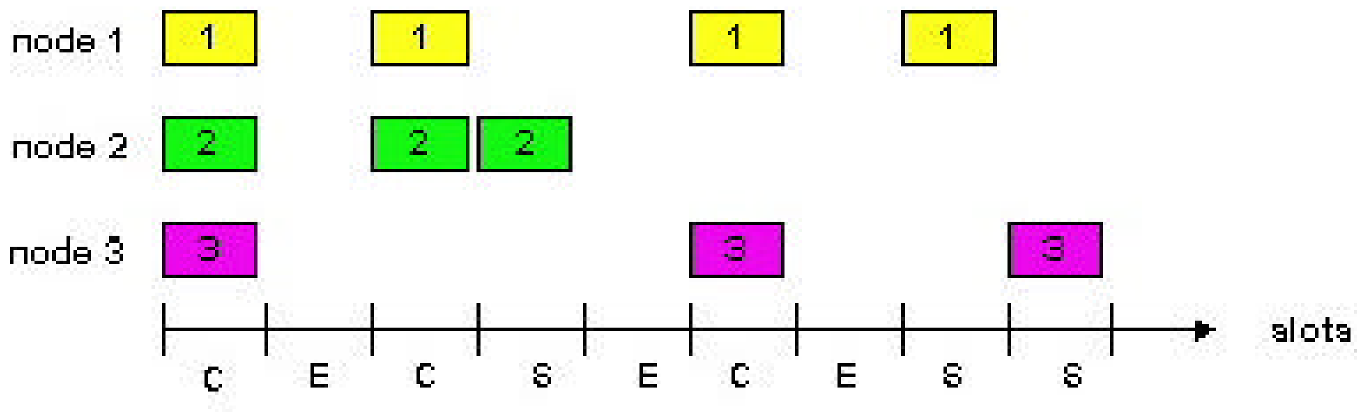
Assumptions

- all frames same size
- time 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

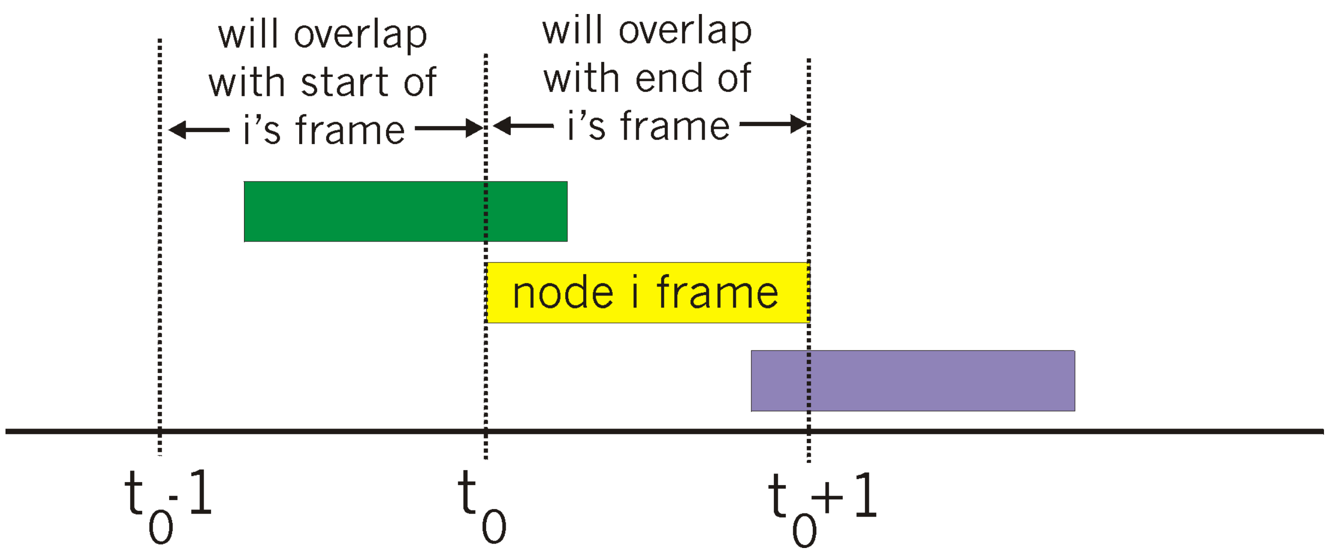
Efficiency is the long-run fraction of successful slots when there's many nodes, each with many frames to send

- 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 $[t_0-1, t_0+1]$



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)

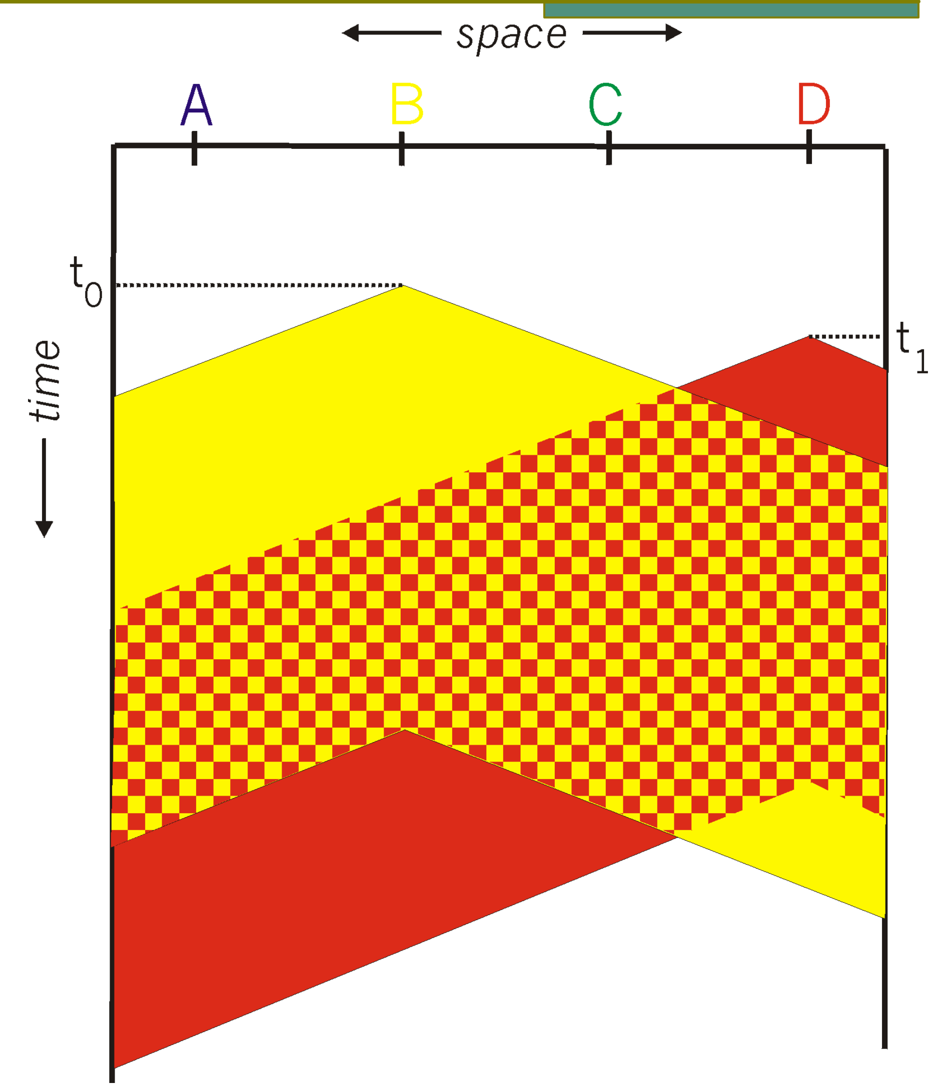
CSMA: listen before transmit:

- If channel sensed idle: transmit entire frame
- If channel sensed busy, defer transmission
  - Persistent CSMA: retry immediately with probability p when channel becomes idle
  - Non-persistent CSMA: retry after a random time interval

Human analogy: don't interrupt others!

CSMA collisions

spatial layout of nodes



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

collision:
entire packet transmission time wasted

note:
1. role of distance & propagation delay in determining collision probability
2. Attenuation and distance

Collision Detection

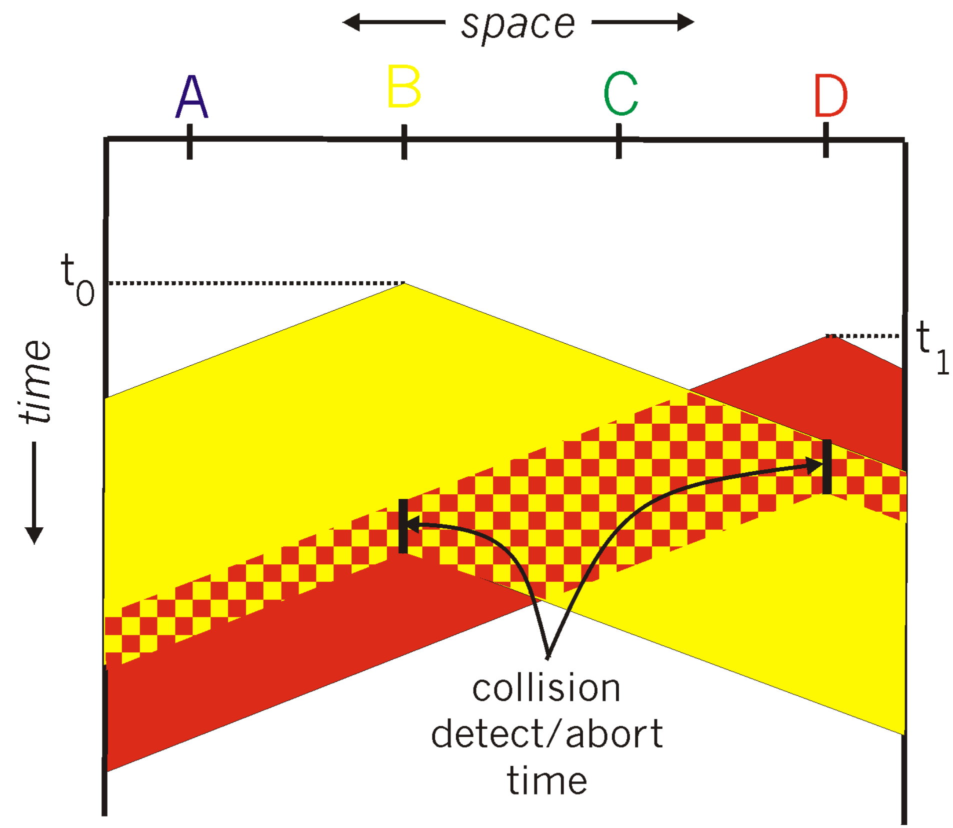
CSMA/CD: carrier sensing, deferral as in CSMA

- 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

- CD circuit operates by looking for voltage exceeding a transmitted voltage
- Want to ensure that a station does not complete transmission prior to 1st bit arriving at farthest-away station
- Time to CD can thus take up to 2x{max prop. delay}

CSMA/CD collision detection



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

LAN Addresses and ARP

32-bit IP address:

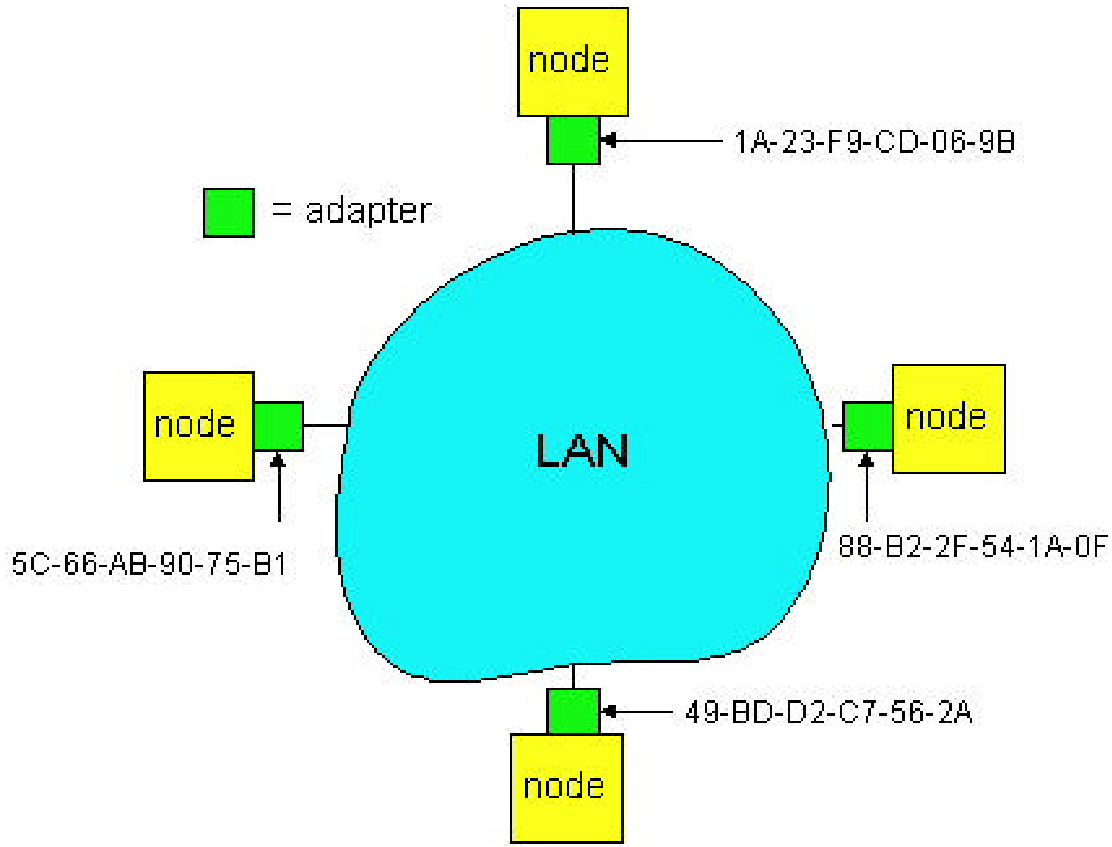
- *network-layer* address
- used to get datagram to destination IP network (recall IP network definition)

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



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

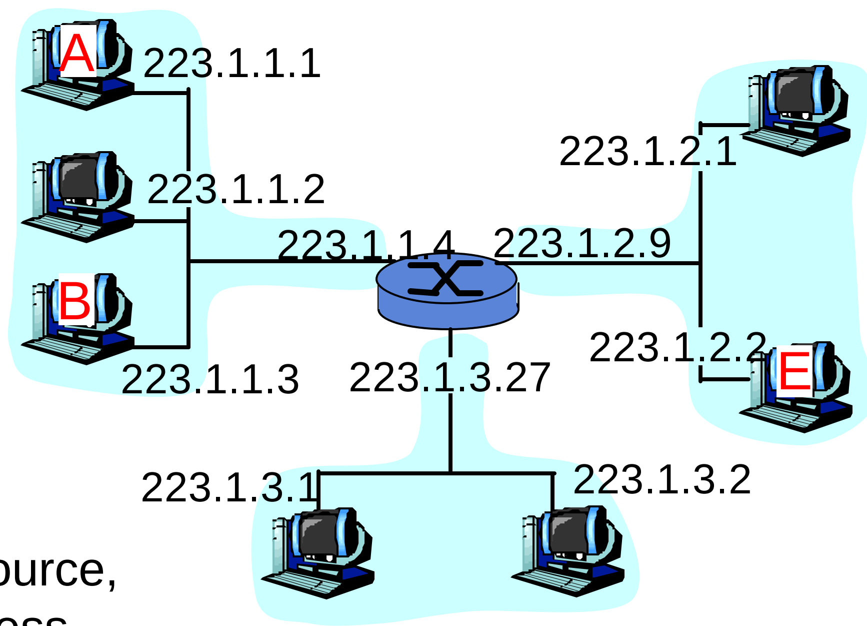
- MAC flat address => portability
  - can move LAN card from one LAN to another

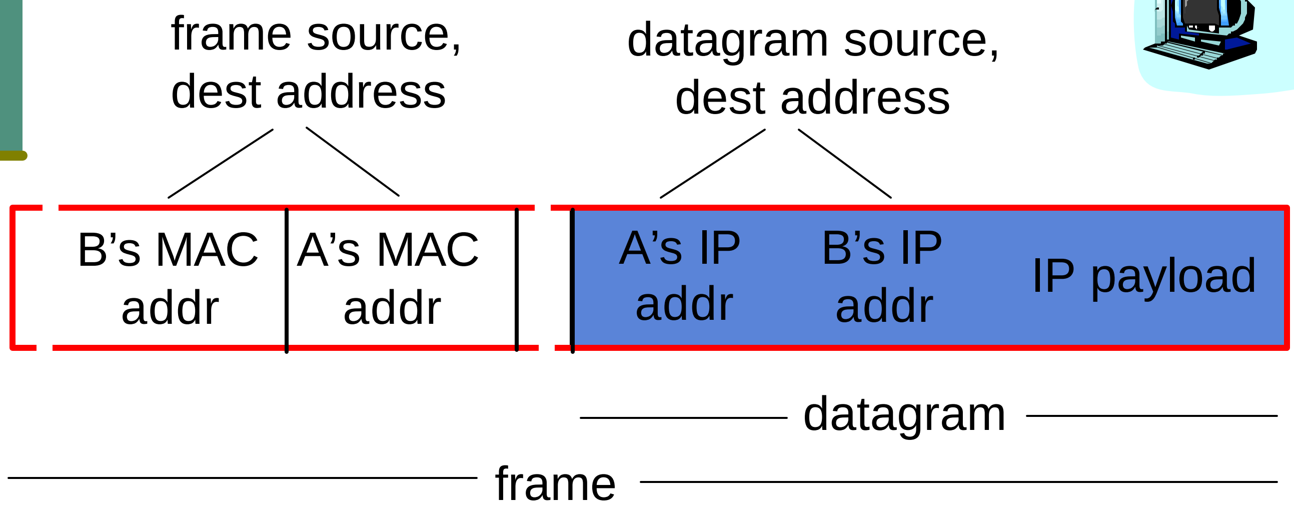
- IP hierarchical address NOT portable
  - depends on IP network to which node is attached

Example

Starting at A, given IP datagram addressed to B:

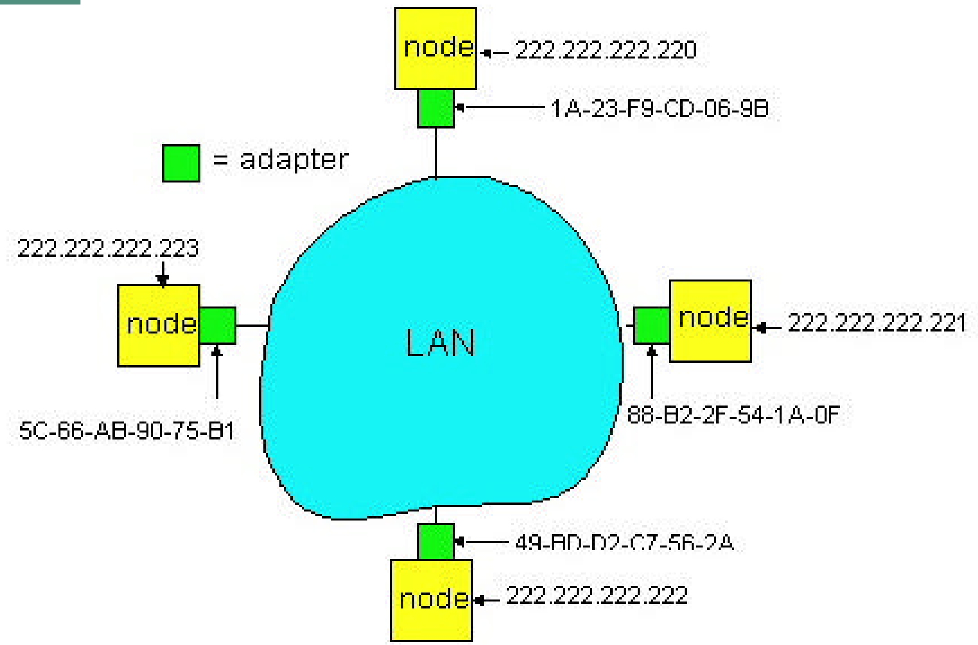
- look up net. address of B, find B on same net. as A
- link layer send datagram to B inside link-layer frame





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)

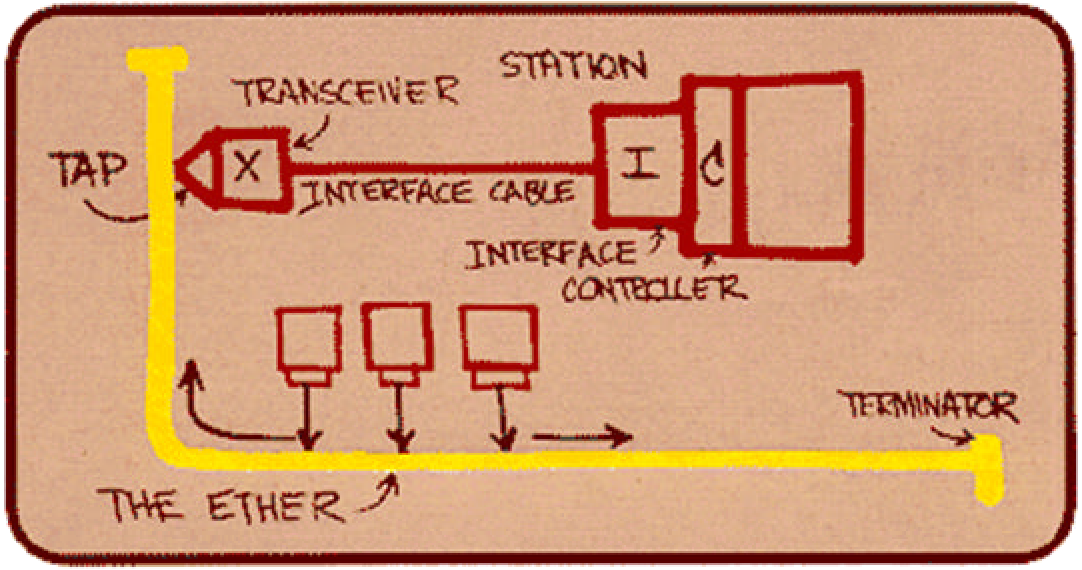
ARP protocol

- 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

Ethernet

"dominant" LAN technology:

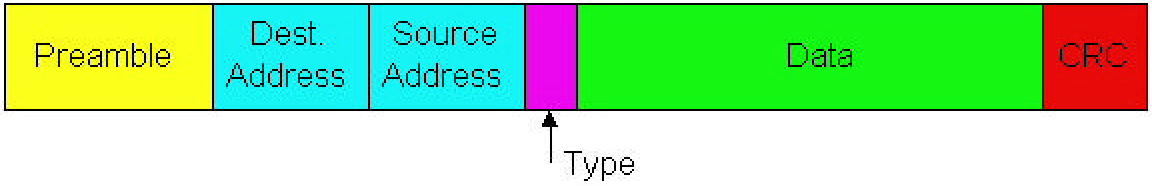
- cheap $20 for 100Mbs!
- first widely used LAN technology
- Simpler, cheaper than token LANs and ATM
- Kept up with speed race: 10, 100, 1000 Mbps



Metcalfe's Ethernet Sketch (1976)

Ethernet Frame Structure

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

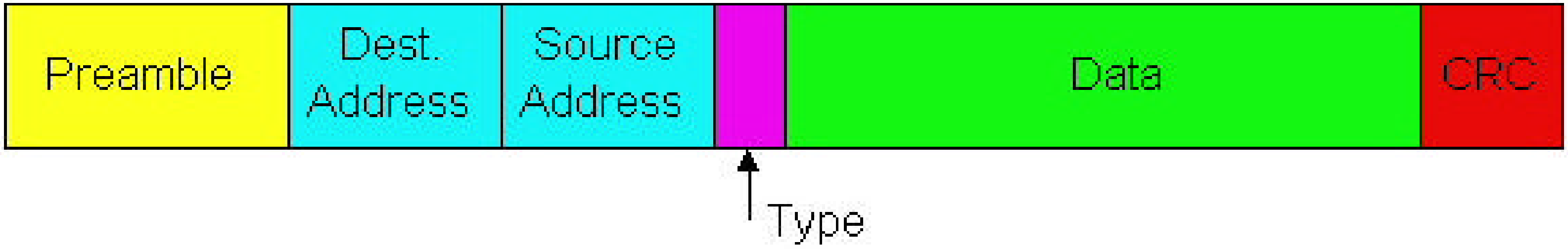


Preamble:

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

Ethernet Frame Structure (more)

- Addresses: 6 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



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

Capture Effect

- Given two stations A & B, an unfair strategy can cause A to continue to "win"
- Assume A & B always ready to send:
  - if busy, both wait, send and collide
  - suppose A wins, B backs off
  - next time, B's chances of winning are halved

Frame Size

- **Minimum:**
  - With repeaters, etc. 802.3 requires 51 µs RTT, corresponding to 512 bit-times
  - Therefore, the minimum frame size is 512 bits (64 Kbytes), which is also called *slot time*
- **Maximum:**
  - 1500 byte limitation on maximum frame transmission size (MTU)
  - Limits maximum buffers at receiver
  - Requires 96 bit Inter-Packet-Gap (IPG)

Ethernet's CSMA/CD (more)

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

Bit time: 0.1 microsec for 10 Mbps Ethernet ;

for K=1023, wait time is about 50 msec

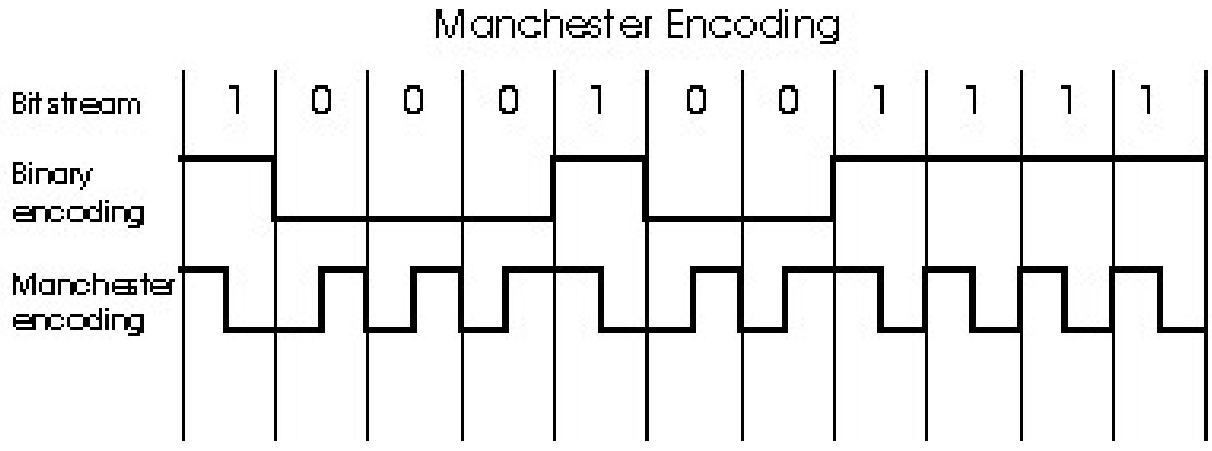
Exponential Backoff:

- *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}

Names

- Structure: [rate][modulation][media or distance]
  - 10Base5 (10Mbps, baseband, coax, 500m)
  - 10Base-T (10Mbps, baseband, twisted pair)
  - 100Base-TX (100Mbps, baseband, 2 pair)
  - 100Base-FX (100Mbps, baseband, fiber)
  - 1000Base-CX for two pairs balanced copper cabling
  - 1000Base-LX for long wavelength optical transmission
  - 1000Base-SX for short wavelength optical transmission
- Wireless (WiFi)
  - 802.11
  - Versions: a, b, g

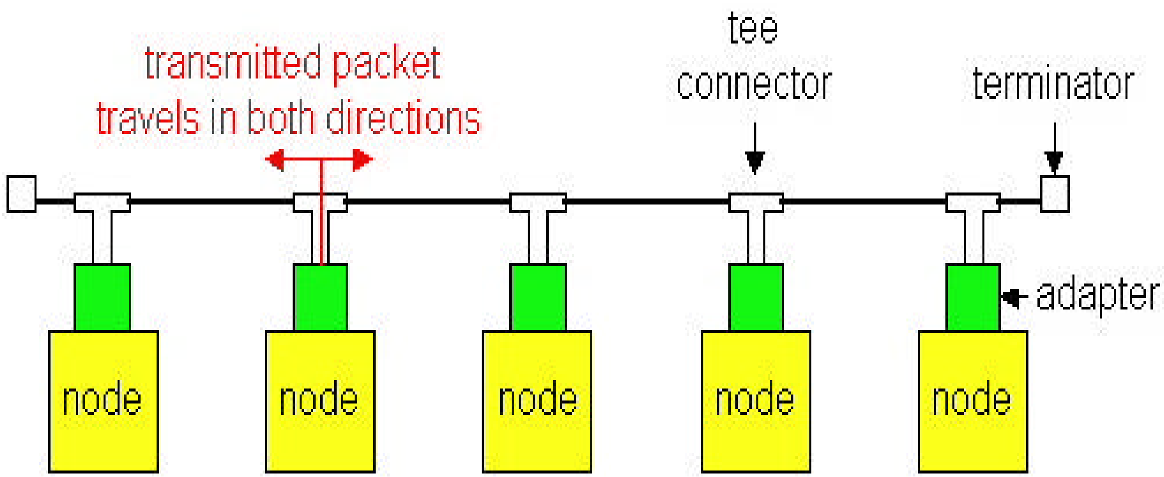
Manchester encoding



- Used in 10BaseT, 10Base2
- Each bit has a transition
- Allows clocks in sending and receiving nodes to synchronize to each other
  - no need for a centralized, global clock among nodes!
- Physical-layer stuff!

Ethernet Technologies: 10Base2

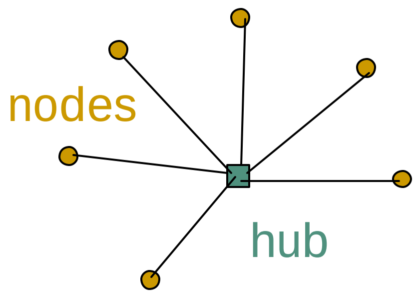
- 10: 10Mbps; 2: under 200 meters max cable length
- thin coaxial cable in a bus topology



- repeaters used to connect up to 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



- 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

Improvements

- Fast Ethernet (1995) adds:
  - 10x speed increase (100m max cable length retains min 64 byte frames)
  - replace Manchester with 4B/5B (from FDDI)
  - full-duplex operation using switches
  - speed & duplex auto-negotiation

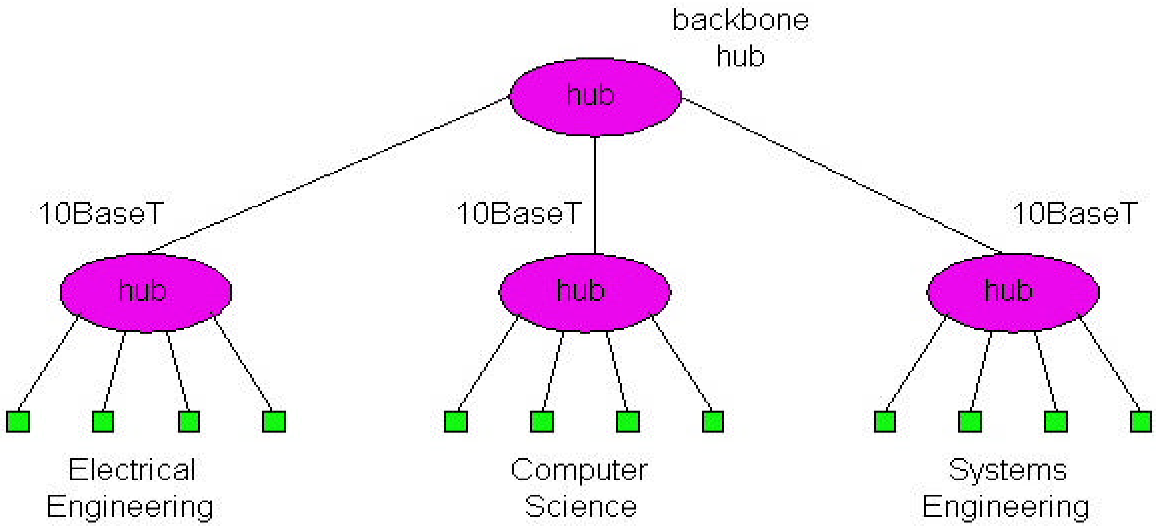
- Gigabit Ethernet IEEE 802.3{z,ab} (1998,9) adds:
  - 1000 Mb/s
  - 100x speed increase
  - carrier extension (invisible padding...)
  - packet bursting

Interconnecting LAN segments

- Hubs
- Bridges
- Switches
  - Remark: switches are essentially multi-port bridges.
  - What we say about bridges also holds for switches!

Interconnecting with hubs

- Backbone hub interconnects LAN segments
- Extends max distance between nodes
- But individual segment collision domains become one large collision domain
  - if a node in CS and a node EE transmit at same time: collision
- Can’t interconnect 10BaseT & 100BaseT

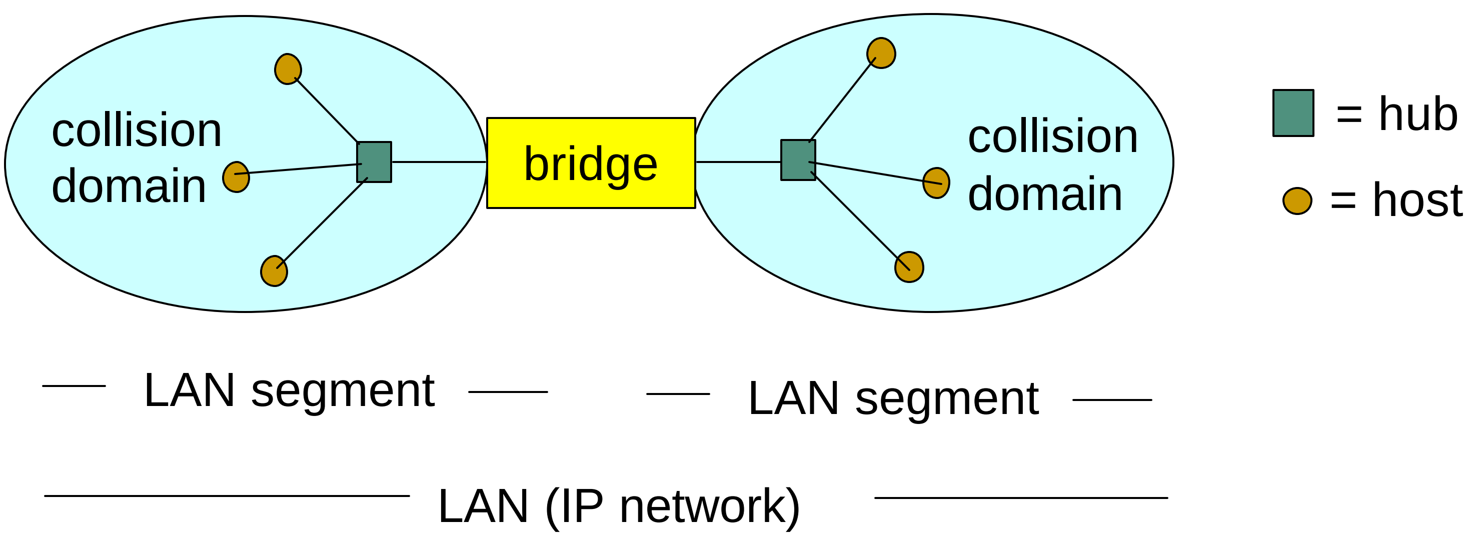


Bridges

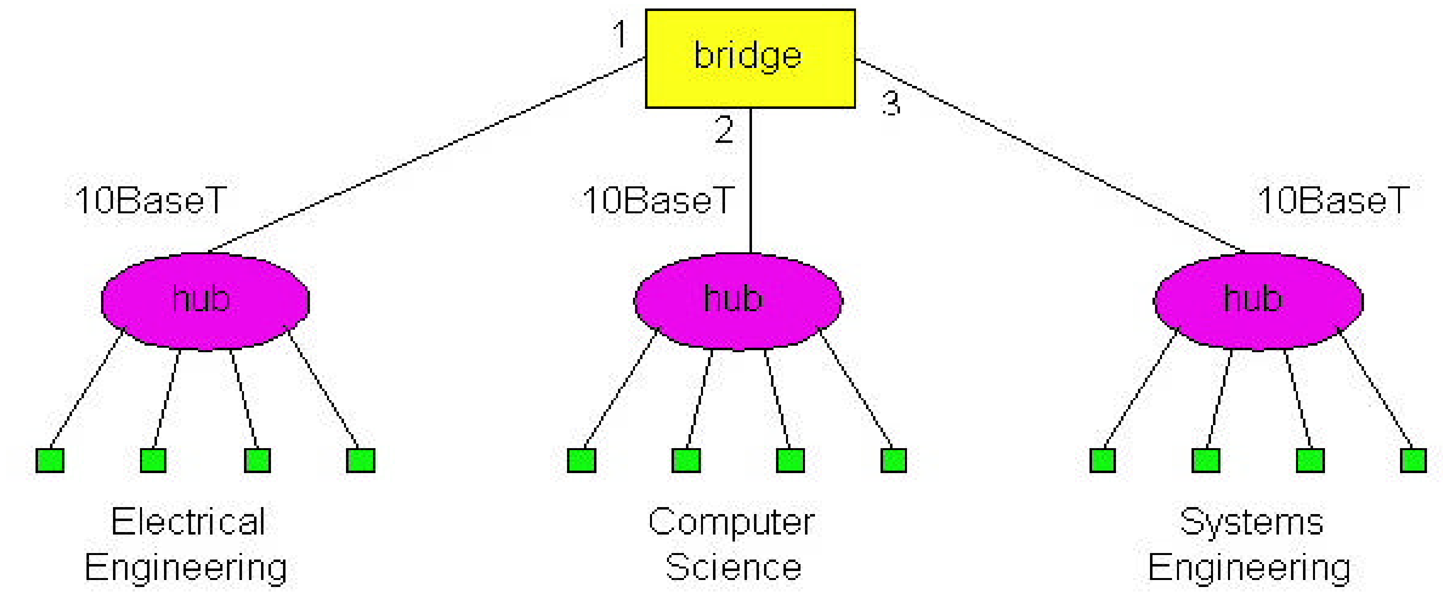
- Link layer device
  - stores and forwards Ethernet frames
  - examines frame header and selectively forwards frame based on MAC dest address
  - when frame is to be forwarded on segment, uses CSMA/CD to access segment
- transparent
  - hosts are unaware of presence of bridges
- plug-and-play, self-learning
  - 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



Forwarding



How do determine to which LAN segment to forward frame?

- Looks like a routing problem...

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

Filtering/Forwarding

<u>When bridge receives a frame:</u>

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

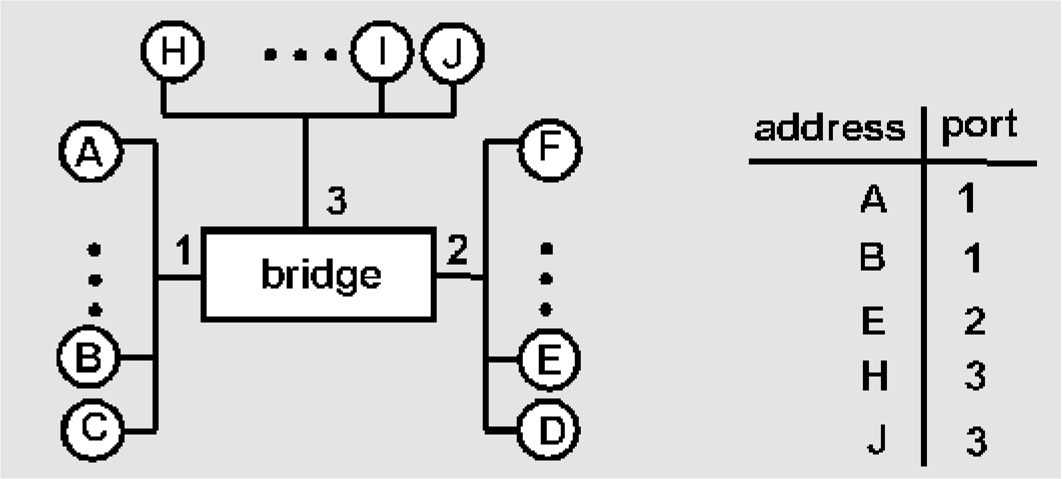
}

else flood

forward on all but the interface on which the frame arrived

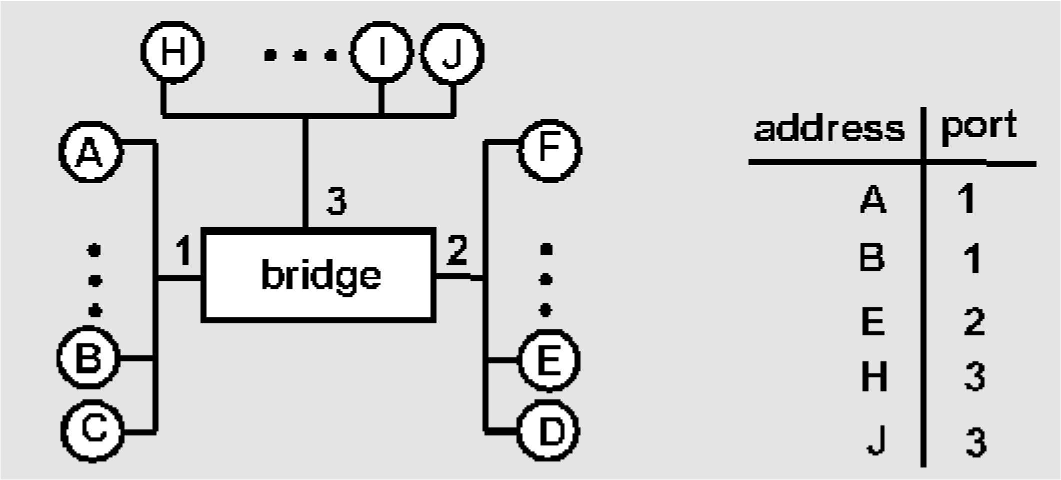
Bridge example

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



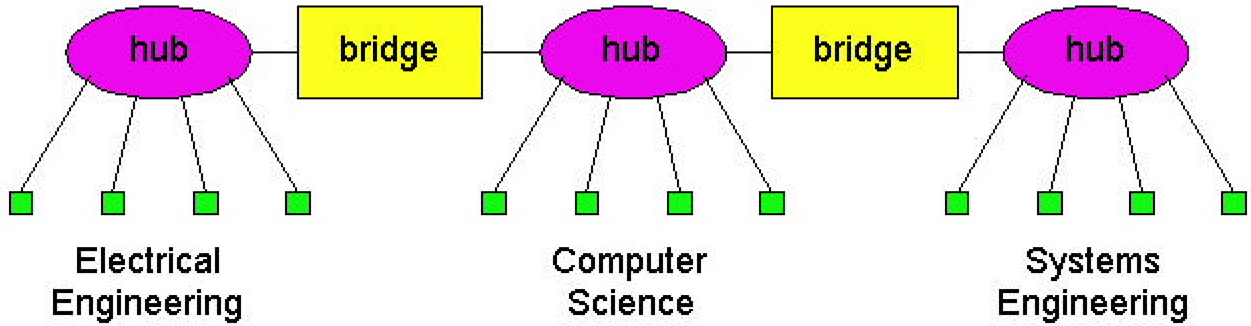
- Bridge receives frame from from C
  - notes in bridge table that C is on interface 1
  - because D is not in table, bridge sends frame into interfaces 2 and 3
- frame received by D

Bridge Learning: example



- 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

Interconnection without backbone

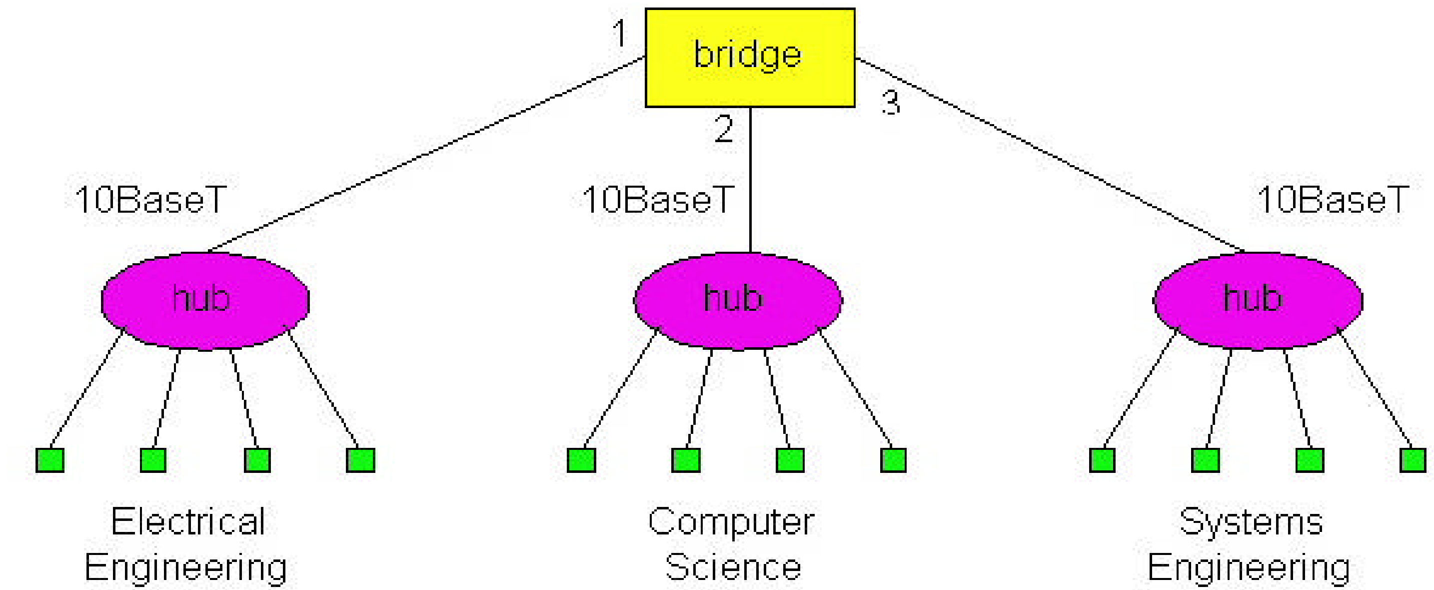


- Not recommended for two reasons:

  - single point of failure at Computer Science hub

  - all traffic between EE and SE must path over CS segment

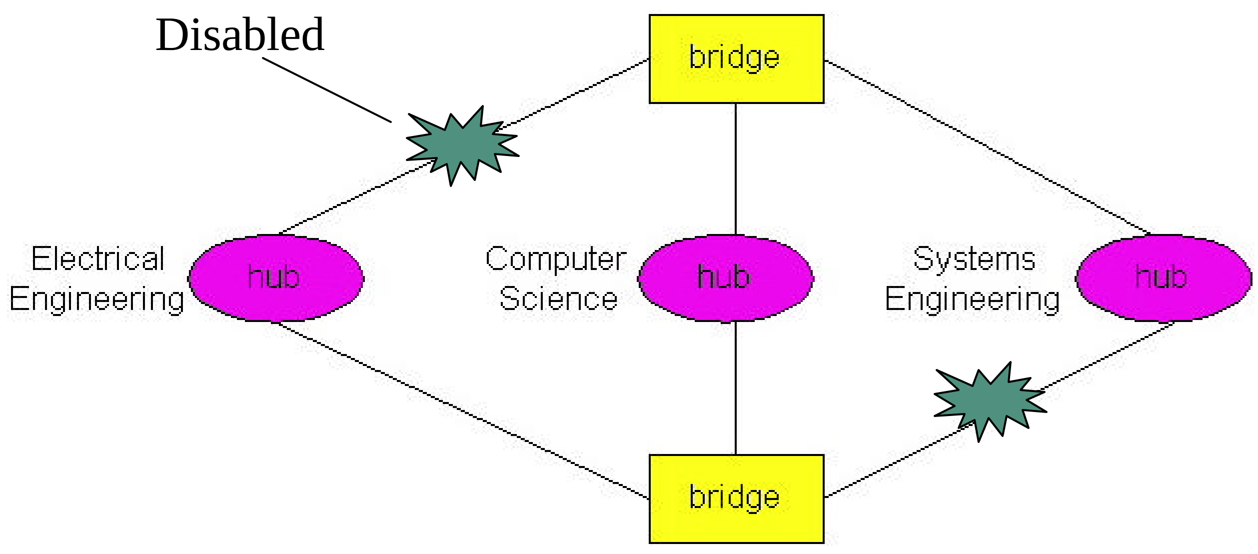
Backbone configuration



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

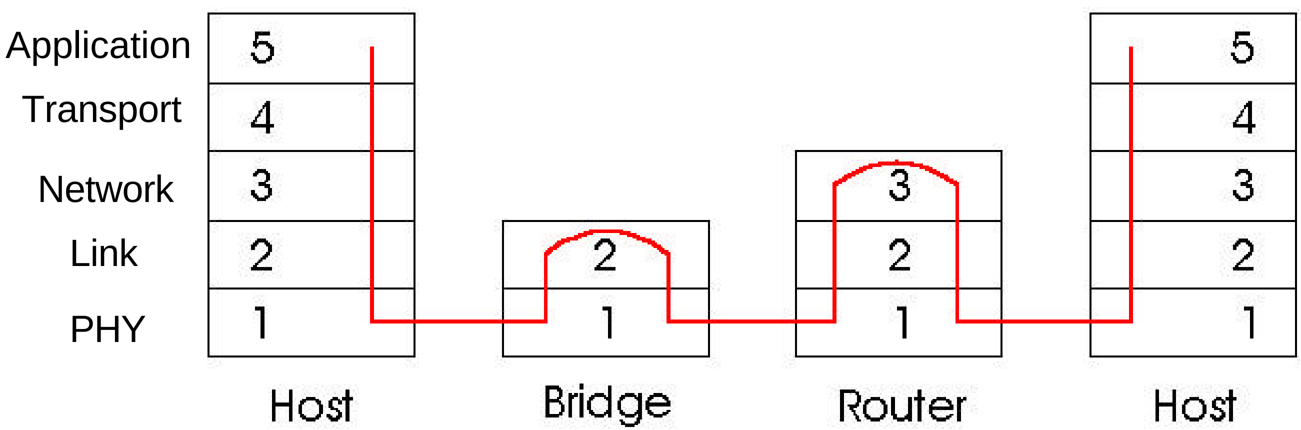


Some bridge features

- 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 are store-and-forward devices
  - routers: network layer devices (examine L-3 headers)
  - bridges are link layer devices (examine L-2 headers)
- Routers maintain routing tables, implement routing algorithms
- Bridges maintain bridge tables, implement learning: filtering, forwarding, and spanning tree algorithms



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

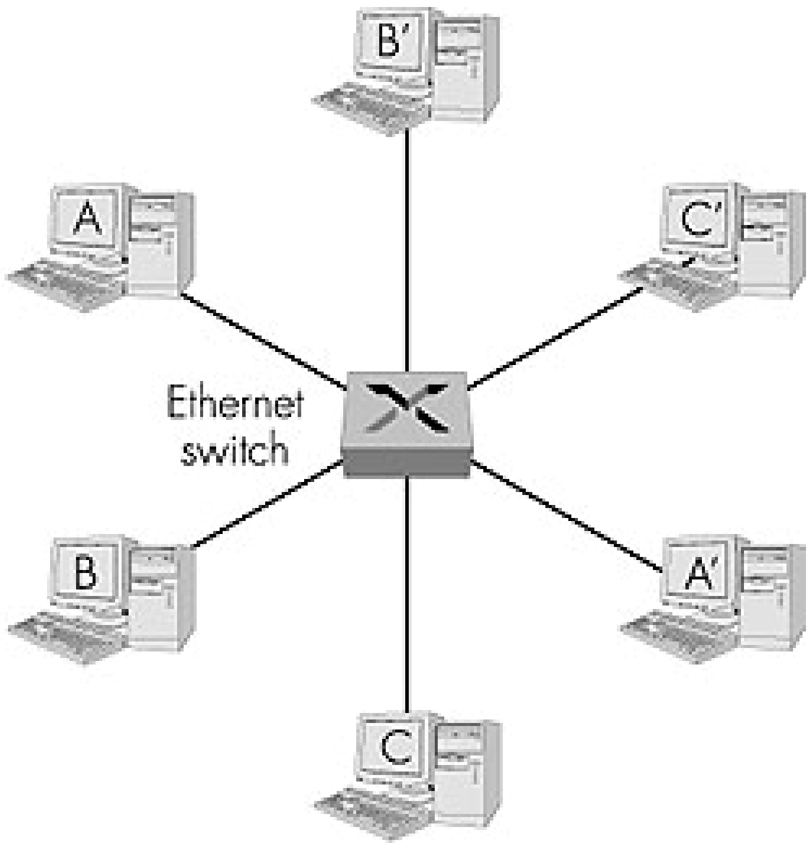
- require IP address configuration (not plug and play)

- 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, and spanning tree.
- Switching: A-to-A' and B-to-B' simultaneously, no collisions
- Large number of interfaces
- Star topology
  - Ethernet, but no collisions!



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

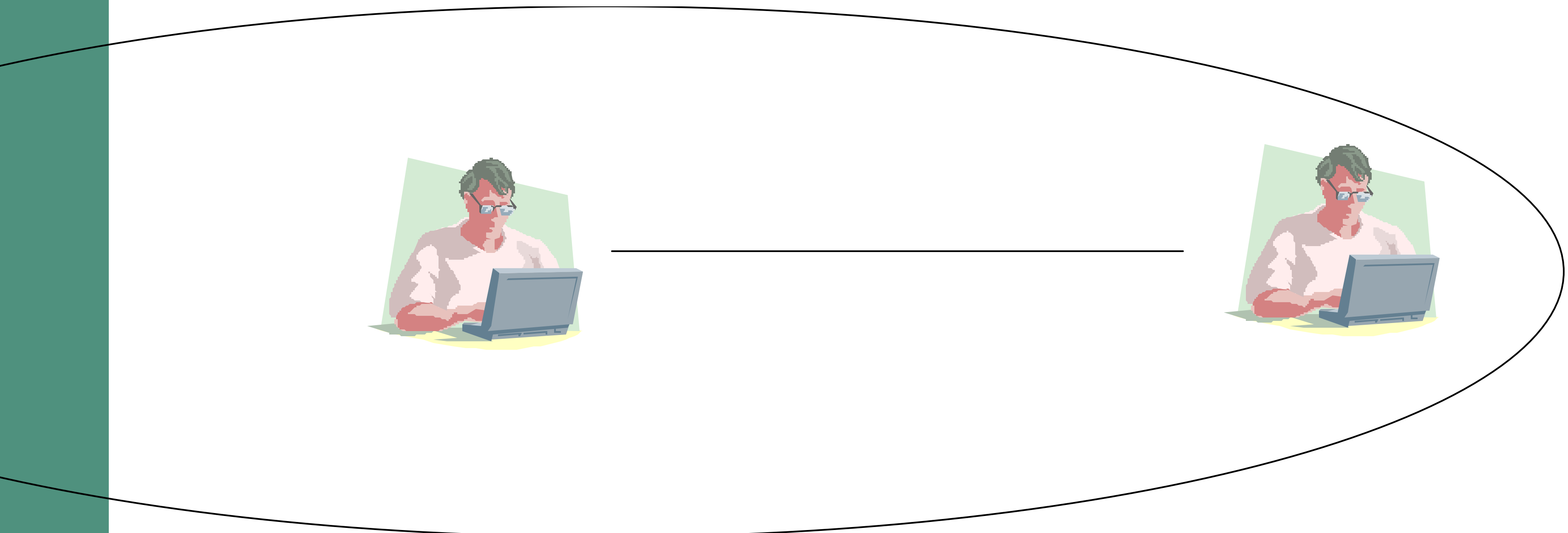
Summary comparison

	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

IEEE 802.11 Wireless LAN (WiFi)

- 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

Wireless LAN Architectures



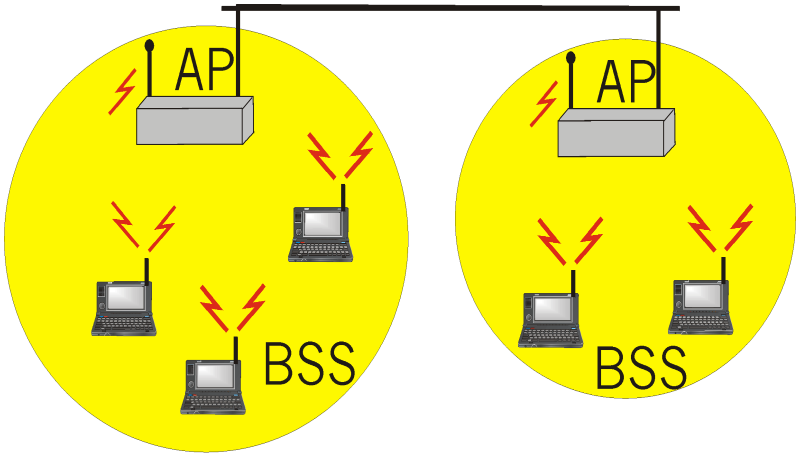
Wireless LANs

Infrastructure based mode
(Base station approach)

Ad Hoc Network
approach

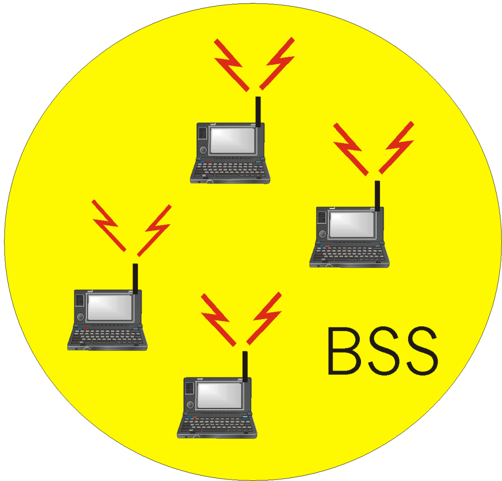
Base station approach

- 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
- BSS’s combined to form distribution system (DS)



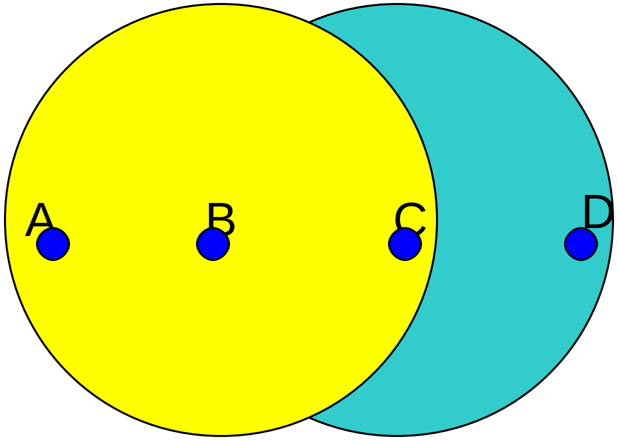
Ad Hoc Network approach

- 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



Wireless: The problems

- Reachability is not transitive



- Hidden nodes: A and C send a packet to B; neither A nor C will detect the collision!
- Exposed node: B sends a packet to A; C hears this and decides not to send a packet to D (despite the fact that this will not cause interference)

IEEE 802.11 MAC Protocol. First: CSMA

802.11 CSMA: sender

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

  then transmit entire frame (no collision detection)

-if sense channel busy

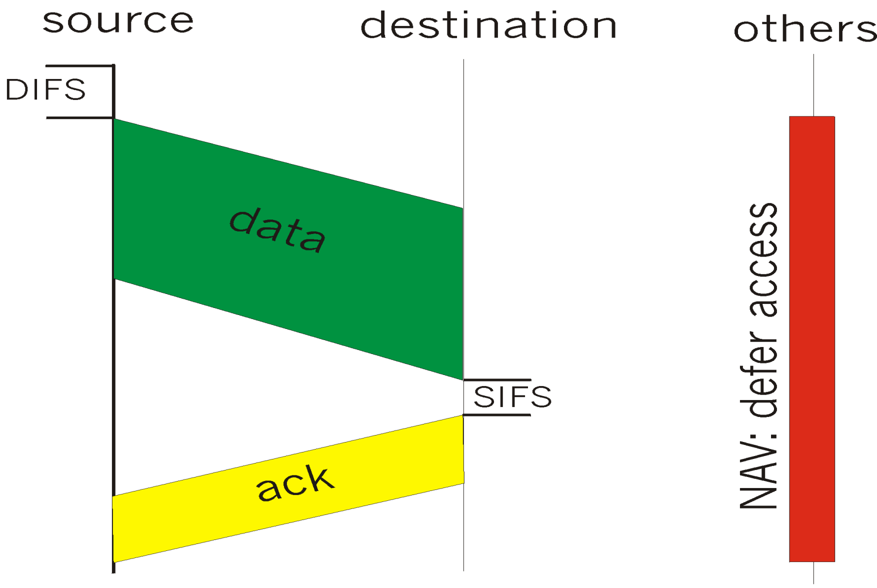
  then binary backoff

802.11 CSMA receiver

- if received OK

  return ACK after **SIFS**

  (ACK is needed due to hidden terminal problem)

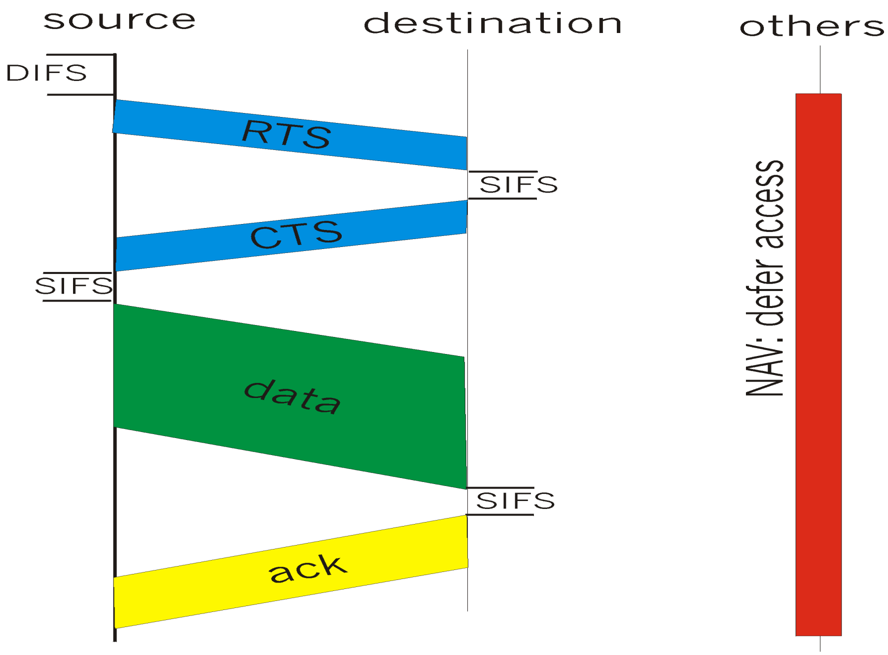


Second: Collision avoidance mechanisms

- 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)

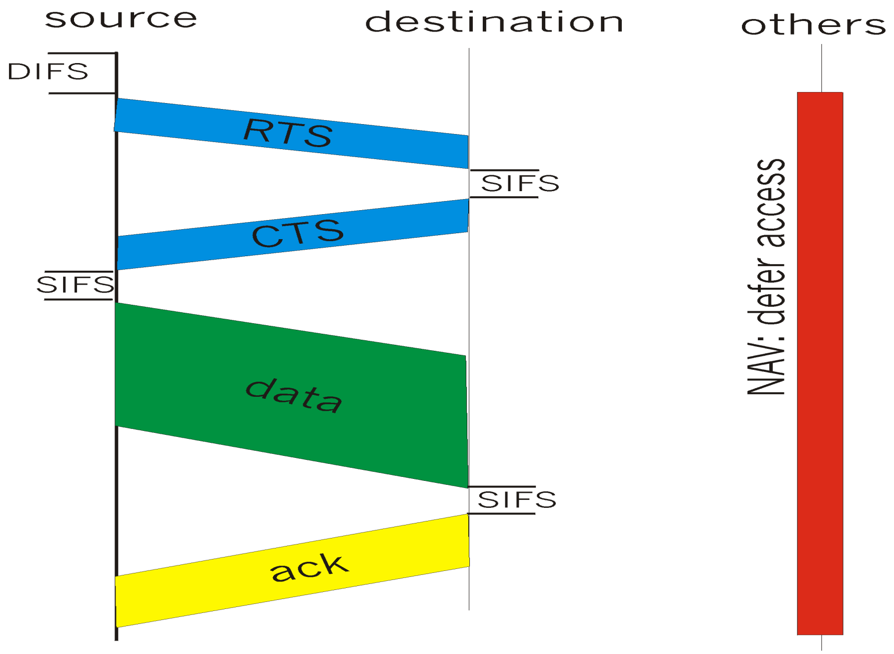
Collision Avoidance: RTS-CTS exchange

- 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



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



CSMA/CA

- CA instead of CD: cannot listen while transmitting
- NAV: Network Allocation Vector: time until channel will become available – transmitted in every frame
- Distributed Coordinated Function: Before transmitting
  - Check if channel is busy, or
  - In an interval between frames [as determined from NAV], or
  - In an interval reserved for retransmission of an erroneous frame [as determined from previous transmission]
  - If one of the above, exponential backoff; else transmit
- Centralized Controlled Access Mechanism: A point coordinator (in access point) coordinates transmissions
  - Stations request that the PC puts them in polling list
  - Periodically, PC polls stations for traffic.

Point-to-Point (PPP)

- One sender, one receiver, one link (full duplex): 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 (LCP)
- Network layer address negotiation: endpoint can learn/configure each other's network address (IPCP)

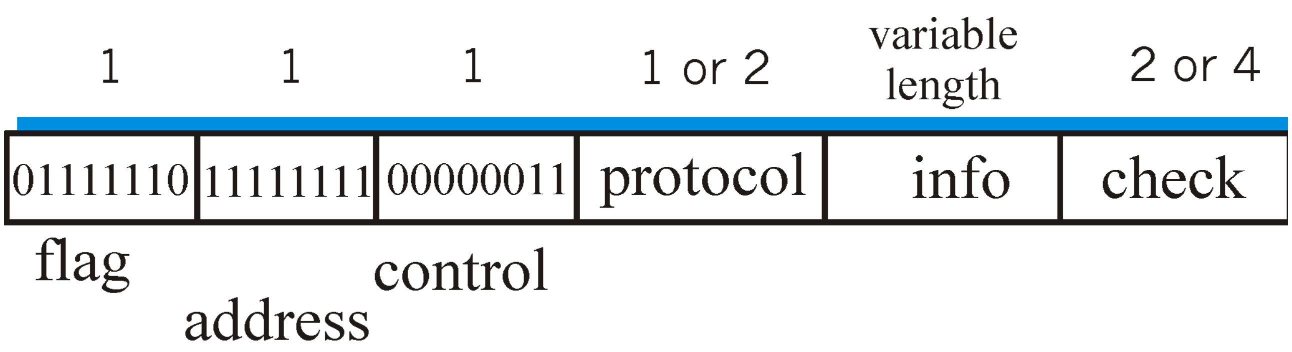
PPP non-requirements

- No error correction/recovery
- No flow control
- Out of order delivery OK
- No need to support multipoint links

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

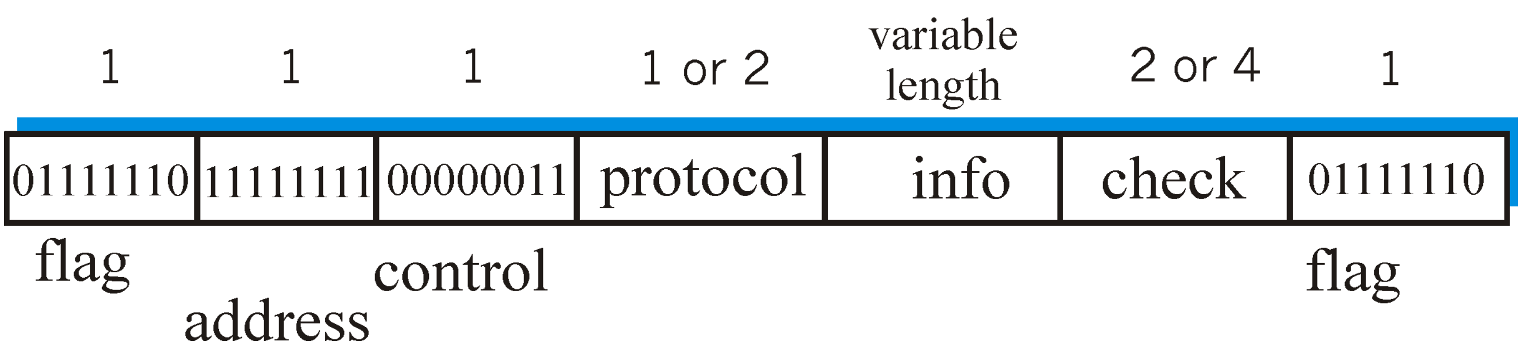
PPP Data Frame

- 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)



PPP Data Frame

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

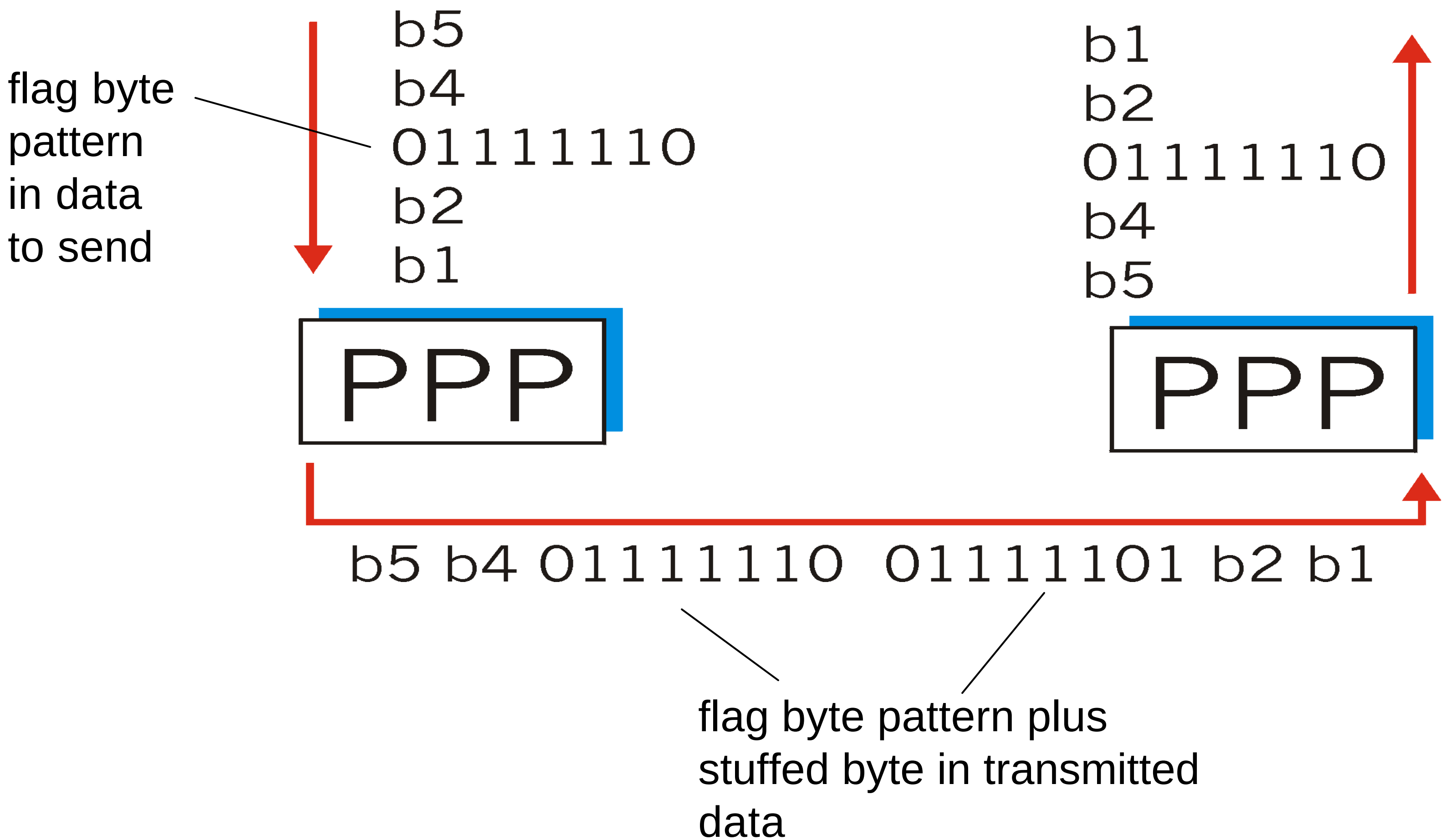


Byte Stuffing

- “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



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