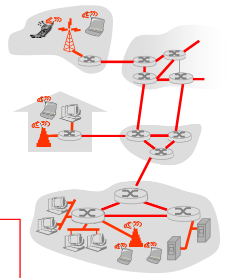
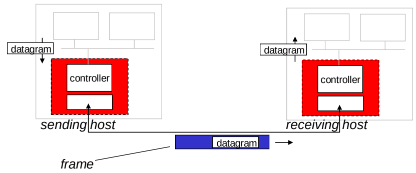
Link Layer: Introduction

- hosts and routers are nodes
- links connect nodes
  - wired links
  - wireless links
- layer-2 packet is a frame, encapsulates datagram



data-link layer transfers datagram from one node to *physically adjacent* node over a link

Adaptors Communicating

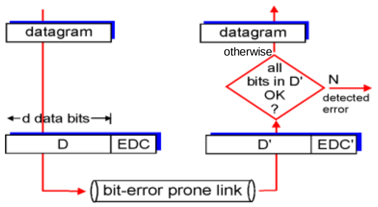


- sender:
  - puts datagram in frame
  - adds error checking, addresses etc.
- receiver
  - looks for errors, etc.
  - extracts datagram, to upper layer

Error Detection

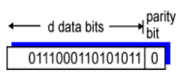
EDC= Error Detection and Correction bits (redundancy)
D = Data protected by error checking, may include header fields

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

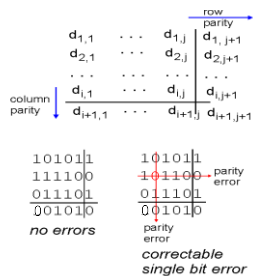


Parity Checking

Single Bit Parity:



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



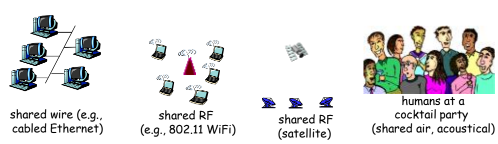
Checksumming: Cyclic Redundancy Check

- Can detect many errors
- More about this on the exercise next week!

Multiple Access Links and Protocols

Two types of "links":

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



Multiple Access protocols

- single shared channel
- collisions possible

Multiple access protocol

- determine when node can transmit
- communication about channel sharing must use channel itself!

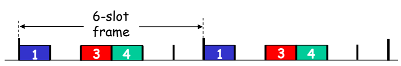
MAC Protocols

- Channel Partitioning
  - divide channel into smaller "pieces" (time slots, frequency)
  - allocate piece to node for exclusive use
- Random Access
  - channel not divided, allow collisions
  - "recover" from collisions
- "Taking turns"
  - nodes take turns, nodes with more to send can take longer turns

Channel Partitioning MAC protocols: TDMA

TDMA: time division multiple access

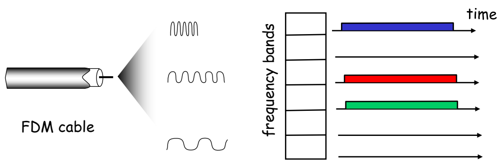
- access to channel in "rounds"
- each sender gets fixed length slot in each round
- unused slots go idle



Channel Partitioning MAC protocols: FDMA

FDMA: frequency division multiple access

- spectrum divided into frequency bands
- each sender assigned fixed frequency band
- unused transmission time in frequency bands go idle



Random Access Protocols

- When node has packet to send
  - transmit at full channel data rate
  - no coordination among nodes
- two or more transmitting nodes ➜ "collision",
- random access MAC protocol specifies:
  - how to detect collisions
  - how to recover from collisions
- Examples of random access MAC protocols:
  - slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA

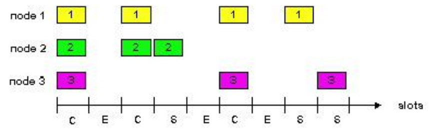
Assumptions:

- all frames same size
- time divided into equal size slots (time to transmit 1 frame)
- nodes start to transmit only slot beginning
- if 2 or more nodes transmit in slot, all nodes detect collision

Operation:

- when node obtains fresh frame, transmits in next slot
  - *if no collision:* node can send new frame in next slot
  - *if collision:* node retransmits frame in each subsequent slot with prob. p until success

Slotted ALOHA

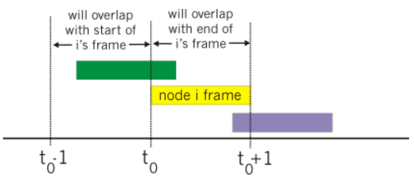


Pros
- single active node can continuously transmit
- decentralized: only slots need to be in sync
- simple

Cons
- collisions, wasting slots
- idle slots
- nodes may be able to detect collision in less than time to transmit packet
- clock synchronization

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 $[t_0-1, t_0+1]$



CSMA (Carrier Sense Multiple Access)

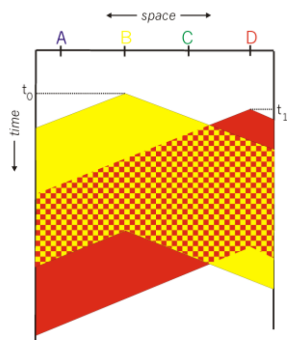
CSMA: listen before transmit:
Channel idle: transmit
Channel busy: wait

Human analogy: don't interrupt others!

CSMA collisions

collisions *can* occur:
caused by propagation delay

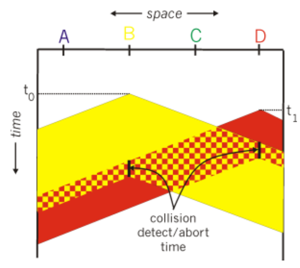
collision:
entire packet transmission time wasted



CSMA/CD (Collision Detection)

- if collision *detected* stop transmission
- collision detection:
  - easy in wired LANs: compare transmitted and received signals
  - difficult in wireless LANs: received signal overwhelmed by local transmission strength
- 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!

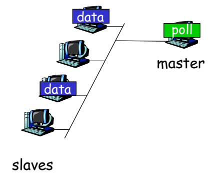
random access MAC protocols
- efficient at low load: single node can fully utilize channel
- high load: collisions

"taking turns" protocols

look for best of both worlds!

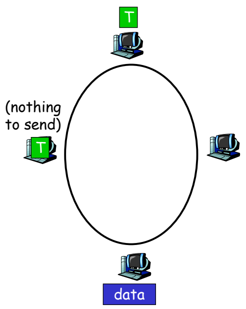
"Taking Turns" MAC protocols

Polling:
- master node "invites" slave nodes to transmit in turn
- disadvantages:
  - polling overhead
  - latency
  - single point of failure (master)



"Taking Turns" MAC protocols

Token passing:
- control **token** passed from one node to next
- disadvantages:
  - token overhead
  - latency
  - single point of failure (token)



Summary of MAC protocols

- *channel partitioning,* by time, frequency or code
  - Time Division, Frequency Division
- *random access* (dynamic),
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - carrier sensing: easy in wired, hard in wireless
  - CSMA/CD used in Ethernet
  - CSMA/CA used in 802.11 (WiFi)
- *taking turns*
  - polling from central site, token passing
  - Bluetooth, IBM Token Ring

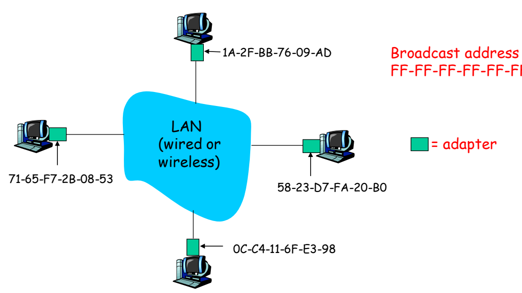
MAC Addresses and ARP

- 32-bit IP address:
  - *network-layer* address
  - used to get datagram to destination IP subnet
- MAC (or LAN or physical or Ethernet) address:
  - function: *get frame from one interface to the next (same network)*
  - 48 bit MAC address (for most LANs)
    - burned in NIC ROM, also sometimes software settable

LAN Addresses and ARP

Each adapter on LAN has unique LAN address

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

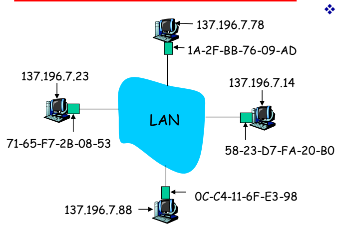


LAN Address (more)

- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space
- analogy:
  - (a) MAC address: like Social Security Number
  - (b) IP address: like postal address
- MAC address does not change
- IP hierarchical address NOT portable
  - address depends on IP subnet to which node is attached

ARP: Address Resolution Protocol

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



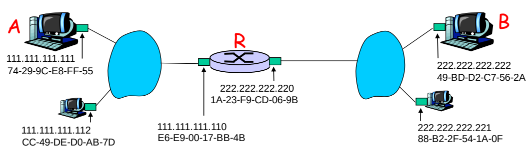
- Each IP node (host, router) on LAN has ARP table
- ARP table: IP/MAC address mappings for some LAN nodes

  < IP address; MAC address; TTL>
  - TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

Addressing: routing to another LAN

send datagram from A to B via R.

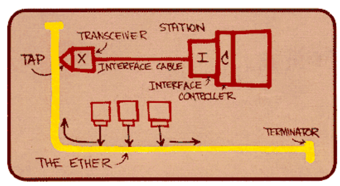
- assume A knows B's IP address
- assume A knows IP address of first hop router, R (how?)



Ethernet

Dominant wired LAN technology:

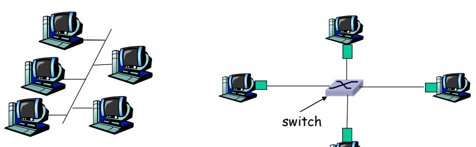
- cheap
- first widely used LAN technology
- simpler, cheaper than token rings



Metcalfe's Ethernet sketch

Star topology

- bus topology popular through mid 90s
  - all nodes can collide with each other
- today: star topology
  - active *switch* in center

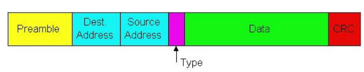


bus: coaxial cable

star

Ethernet Frame Structure

Sending adapter encapsulates IP datagram in Ethernet frame

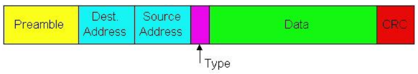


Preamble:

- 7 bytes with pattern 10101010 followed by one byte with pattern 10101011
- synchronizes receiver and sender clock rates

Ethernet Frame Structure (more)

- Addresses: 6 bytes
  - if adapter receives frame with matching destination address, or with broadcast address (e.g. ARP packet), it passes data in frame to network layer protocol
  - otherwise, adapter discards frame
- Type: indicates higher layer protocol (mostly IP)
- CRC: checked at receiver, if error is detected, frame is dropped



Ethernet: Unreliable, connectionless

- connectionless: No handshaking between sending and receiving NICs
- unreliable: receiving NIC doesn't send acks or nacks to sending NIC
  - stream of datagrams passed to network layer can have gaps
  - gaps will be filled if app is using TCP
  - otherwise, app will see gaps
- Ethernet's MAC protocol: CSMA/CD

Ethernet CSMA/CD algorithm

1. NIC receives datagram from network layer, creates frame
2. If NIC senses channel idle, starts frame transmission If NIC senses channel busy, waits until channel idle, then transmits
3. If NIC transmits entire frame without detecting another transmission, NIC is done with frame !
4. If NIC detects another transmission while transmitting, aborts and sends jam signal
5. After aborting, NIC enters exponential backoff: wait random time, after that return to 2.

Ethernet's CSMA/CD (more)

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

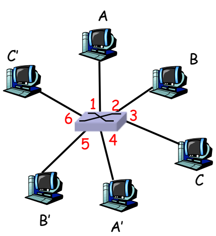
Bit time: .1 microsec for 10 Mbps Ethernet

Exponential Backoff:

- first collision: choose K from {0,1}; delay is K· 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}

Switch: allows *multiple* simultaneous transmissions

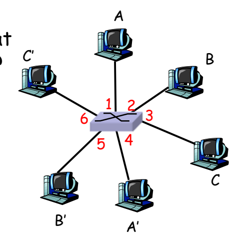
- hosts have dedicated, direct connection to switch
- switches buffer packets
- Ethernet protocol used on *each* incoming link, but no collisions
- *switching:* A-to-A' and B-to-B' simultaneously, without collisions



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

Switch Table

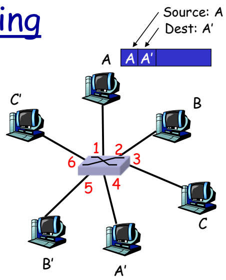
- *Q:* how does switch know that A' reachable via interface 4?
- *A:* each switch has a switch table
- looks like a routing table!
- *Q:* how are entries created, maintained in switch table?



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

Switch: self-learning

- switch *learns* which hosts can be reached through which interfaces
  - when frame received, switch "learns" location of sender: incoming LAN segment
  - records sender/location pair in switch table

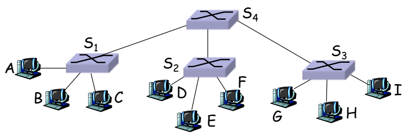


MAC addr	interface	TTL
A	*1*	*60*

Switch table (initially empty)

Interconnecting switches

- switches can be connected together



- Sending from A to G - how does S_1 know to forward frame destined to F via S_4 and S_3?