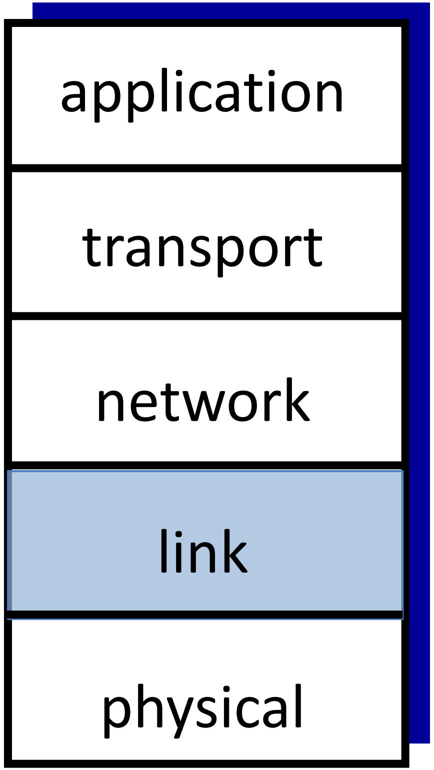
CS 4390: Link Layer

Shuang Hao
University of Texas at Dallas
Fall 2023

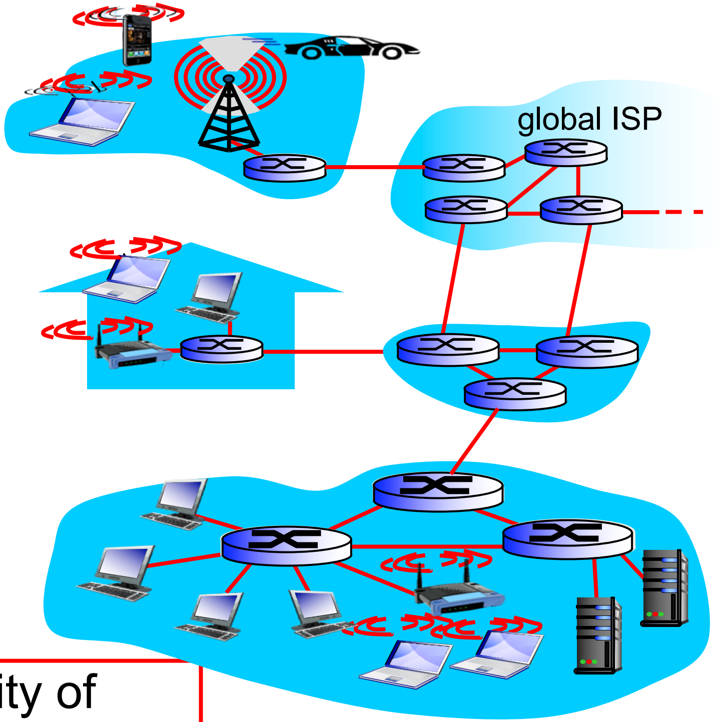
Link Layer

- Our goals
  - understand principles behind link layer services:
    - error detection, correction
    - sharing a broadcast channel: multiple access
    - link layer addressing
    - local area networks: Ethernet, VLANs
  - instantiation, implementation of various link layer technologies



Link Layer: Introduction

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



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

Link Layer Services

- framing, link access:
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - "MAC" addresses used in frame headers to identify source, dest
    - different from IP address!
- reliable delivery between adjacent nodes
  - we learned how to do this already (on transport layer)
  - seldom used on low bit-error link (fiber, some twisted pair)
  - wireless links: high error rates

Link Layer Services

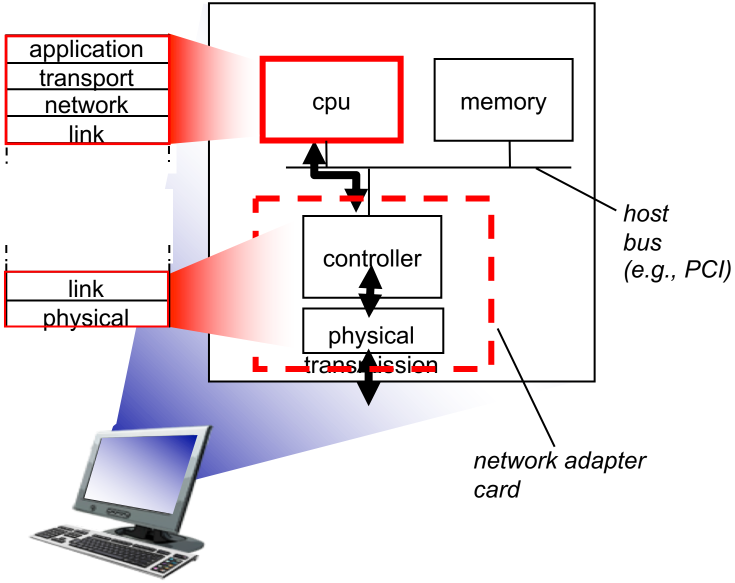
- framing, link access:
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - "MAC" addresses used in frame headers to identify source, dest
    - different from IP address!
- reliable delivery between adjacent nodes
  - we learned how to do this already (on transport layer)
  - seldom used on low bit-error link (fiber, some twisted pair)
  - wireless links: high error rates

Link Layer Services

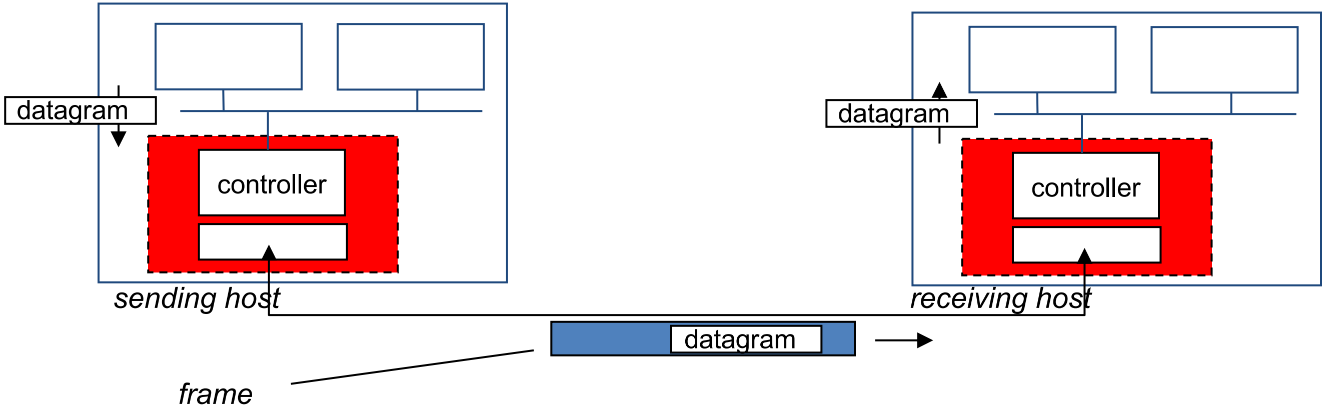
- Flow control:
  - pacing between adjacent sending and receiving nodes
- Error detection:
  - errors caused by signal weakening, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
- Error correction:
  - receiver identifies and corrects bit error(s) 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

Where Is the Link Layer Implemented?

- In each and every host
- Link layer implemented in “adaptor” (aka network interface card NIC) or on a chip
  - Ethernet card, 802.11 card; Ethernet chipset
  - implements link, physical layer
- Attaches into host’s system buses
- Combination of hardware, software, firmware



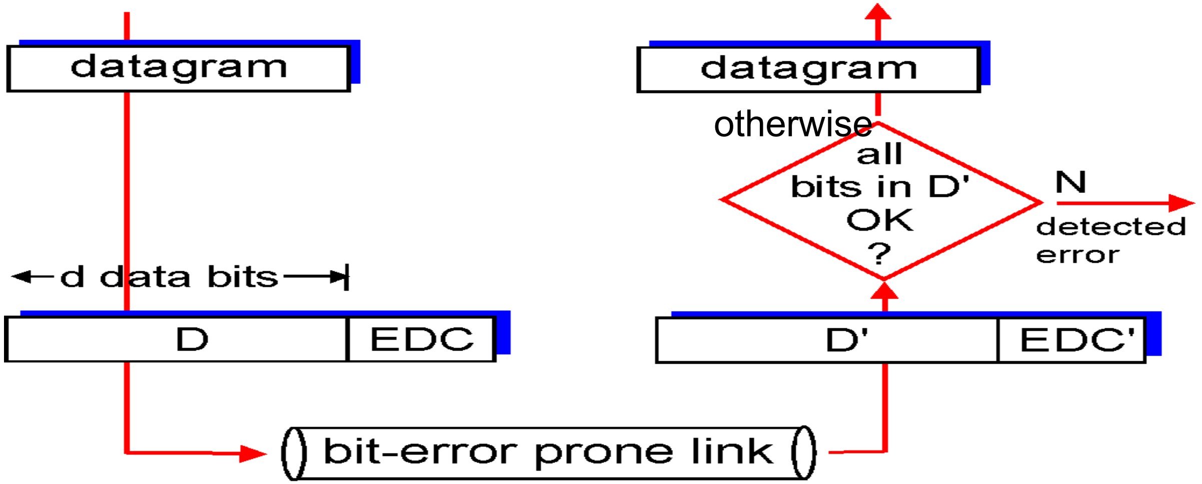
Adaptors Communicating



- Sending side:
  - encapsulates datagram in frame
  - adds error checking bits, reliable data transfer, flow control, etc.
- Receiving side
  - looks for errors, reliable data transfer, flow control, etc
  - extracts datagram, passes to upper layer at receiving side

Error Detection

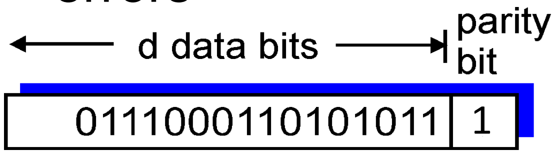
- EDC= Error Detection and Correction bits (redundancy)
  - D = Data protected by error checking, may include header fields
- Error detection not 100% reliable
  - larger EDC field yields better detection and correction



Parity Checking

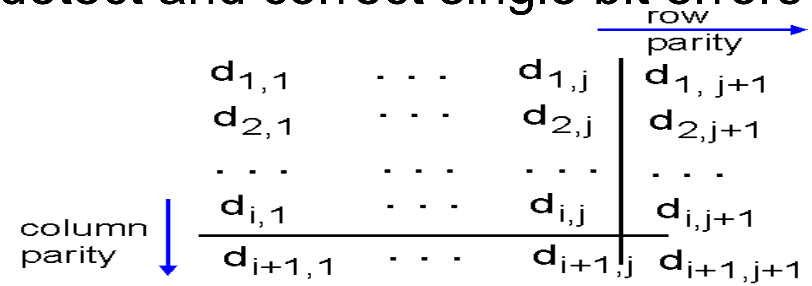
single bit parity:

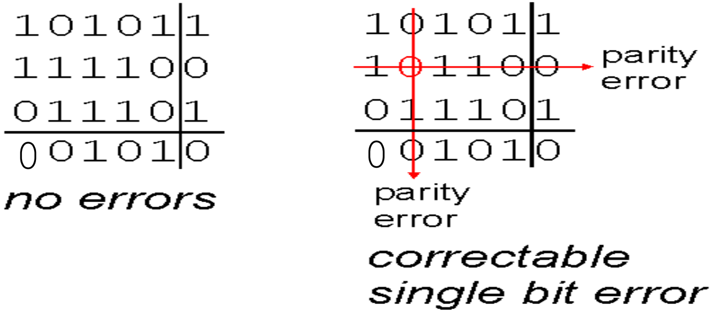
- detect single bit errors



two-dimensional bit parity:

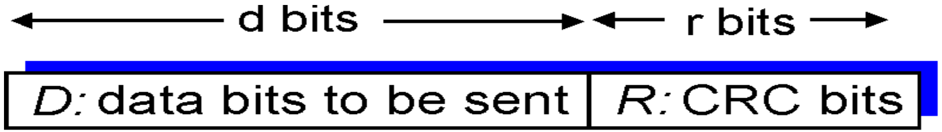
- detect and correct single bit errors





Cyclic Redundancy Check

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



bit pattern

$D * 2^r$ XOR R

mathematical formula

CRC Example

want:

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

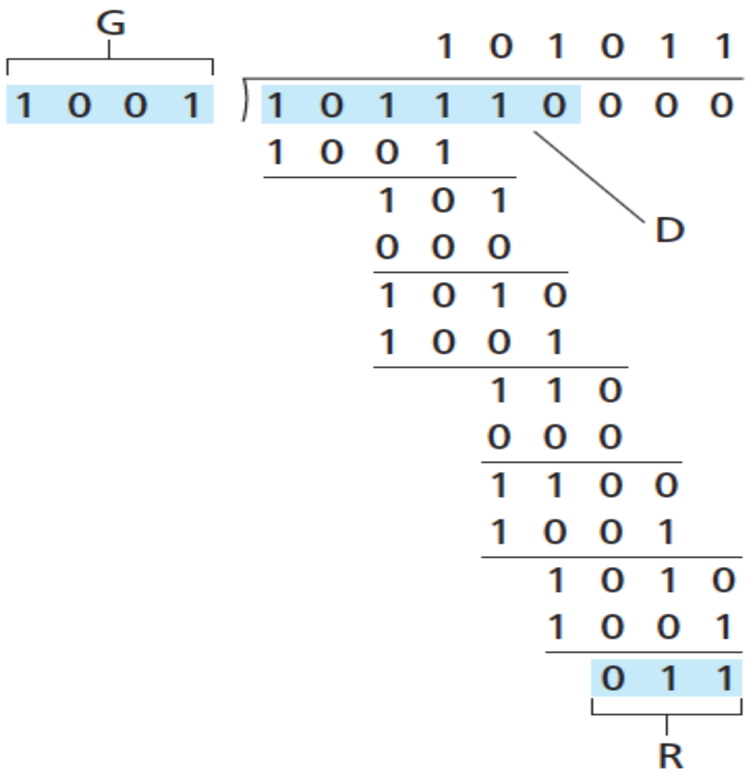
equivalently:

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

equivalently:

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

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



Recap Where We're At

- Link layer
  - Link layer services
  - Error detection

Outline

- MAC protocols
  - Channel partitioning
  - Random access with collision (CSMA)

Multiple Access Protocols

- Single shared broadcast channel
  - two or more simultaneous transmissions by nodes: interference
  - collision if node receives two or more signals at the same time

- Multiple access protocol
  - Determines how nodes share channel, i.e., determine when node can transmit
  - Communication about channel sharing must use channel itself
    - No out-of-band channel for coordination

An Ideal Multiple Access Protocol

Given: broadcast channel of rate R bps

Goal:

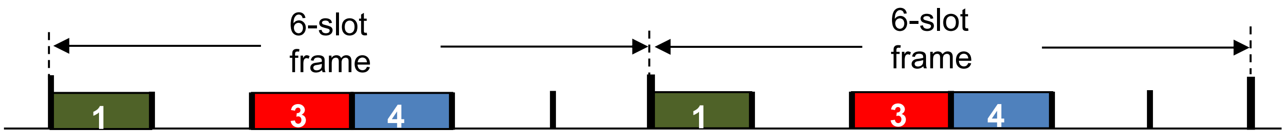
1. when one node wants to transmit, it can send at rate R.
2. when M nodes want to transmit, each can send at average rate R/M
3. fully decentralized:
   - no special node to coordinate transmissions
   - no synchronization of clocks, slots
4. simple

MAC Protocols: Taxonomy

- MAC (Medium Access Control)
- Three broad classes:
  - Channel partitioning
    - divide channel into smaller “pieces” (time slots, frequency, code)
    - allocate piece to node for exclusive use
  - Random access
    - channel not divided, allow collisions
    - “recover” from collisions
  - “Taking turns”
    - nodes take turns, but nodes with more to send can take longer turns

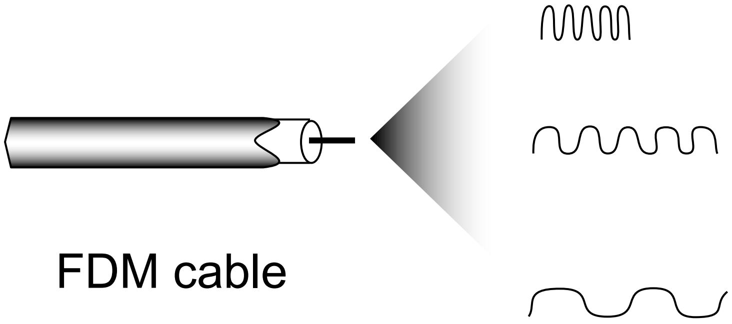
Channel Partitioning MAC protocols: TDMA

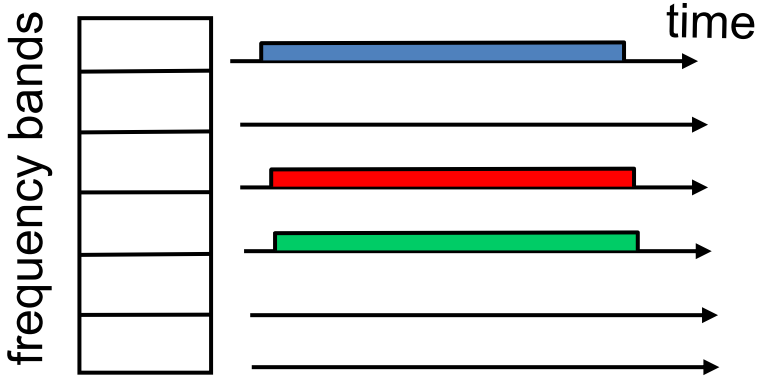
- TDMA: time division multiple access
  - access to channel in "rounds"
  - each station gets fixed length slot (length = frame trans time) in each round
  - unused slots go idle
  - example: 6-station LAN, 1,3,4 have frame, slots 2,5,6 idle



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 data, frequency bands 2,5,6 idle





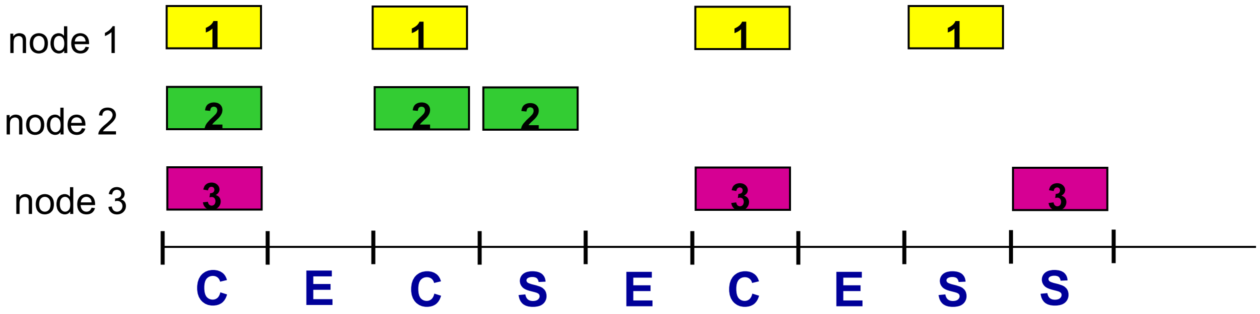
Random Access Protocols

- When node has packet to send
  - transmit at full channel data rate R.
  - no a priori coordination among nodes
- Two or more transmitting nodes → “collision”,
- Random access MAC protocol specifies:
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed retransmissions)
- Examples of random access MAC protocols:
  - slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA

- Assumptions:
  - all frames same size
  - time divided into equal size slots (time to transmit 1 frame)
  - nodes start to transmit only slot beginning
  - nodes are synchronized
  - if 2 or more nodes transmit in slot, all nodes detect collision
- Operation:
  - when node obtains fresh frame, transmits in next slot
    - 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 at full rate of channel
  - highly decentralized: only slots in nodes need to be in sync
  - simple
- Cons:
  - collisions, wasting slots
  - idle slots
  - nodes may be able to detect collision in less than time to transmit packet
  - clock synchronization

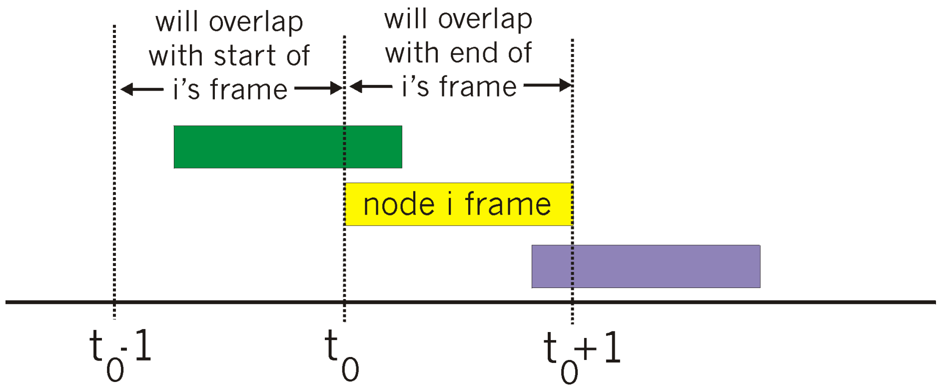
Slotted ALOHA: Efficiency

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

- max efficiency: find p* that maximizes $Np(1-p)^{N-1}$
- for many nodes, take limit of $Np^*(1-p^*)^{N-1}$ as N goes to infinity, gives:
  - max efficiency = 1/e = 0.37
- 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 t0 collides with other frames sent in [t0-1,t0+1]



Pure ALOHA Efficiency

P(success by given node) = P(node transmits) $\cdot$

P(no other node transmits in $[t_0\text{-}1, t_0]$ $\cdot$

P(no other node transmits in $[t_0, t_{0+1}]$

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

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

… choosing optimum p and then letting n to infinity

$$= 1/(2e) = 0.18$$

even *worse* than slotted Aloha

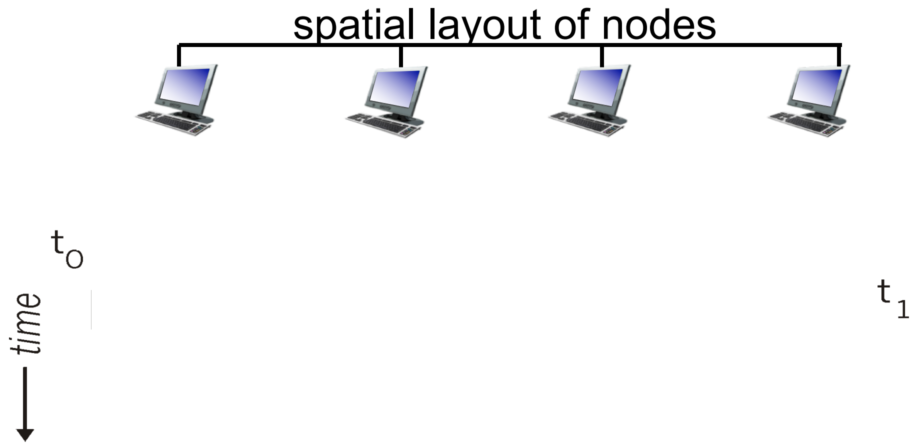
CSMA (Carrier Sense Multiple Access)

- CSMA: listen before transmit:
  - if channel sensed idle: transmit entire frame
  - if channel sensed busy, defer transmission

- human analogy: don't interrupt others

CSMA Collisions

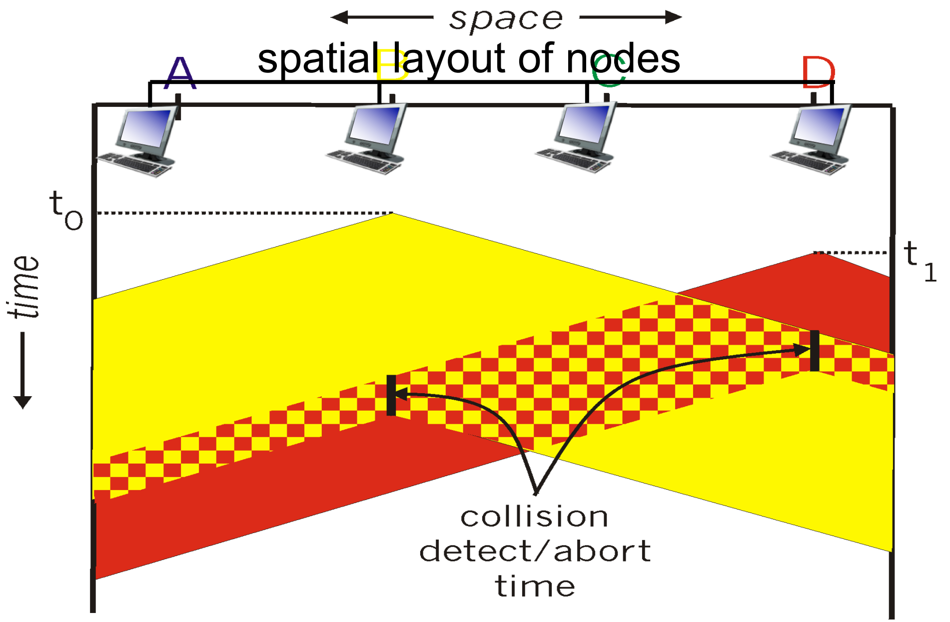
- collisions can still occur: propagation delay means two nodes may not hear each other's transmission
- collision: entire packet transmission time wasted
  - distance & propagation delay play role in in determining collision probability



CSMA/CD (Collision Detection)

- CSMA/CD:
  - collisions detected within short time
  - colliding transmissions aborted, reducing channel wastage
- Collision detection:
  - easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - difficult in wireless LANs: received signal strength overwhelmed by local transmission strength

CSMA/CD (Collision Detection)



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
5. After aborting, NIC enters *binary (exponential) backoff:*
   – after *m*th collision, NIC chooses *K* at random from $\{0,1,2, \ldots, 2^m-1\}$. NIC waits K·512 bit times, returns to Step 2
   – longer backoff interval with more collisions

CSMA/CD Efficiency

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

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

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

Recap Where We're At

- MAC protocols
  - Channel partitioning
  - Random access with collision (CSMA)

Outline

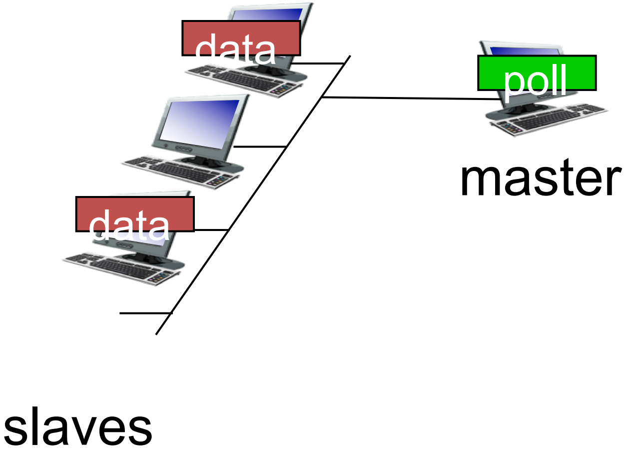
- Taking-turns MAC protocols
- MAC addresses
- ARP (Address Resolution Protocol)

“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

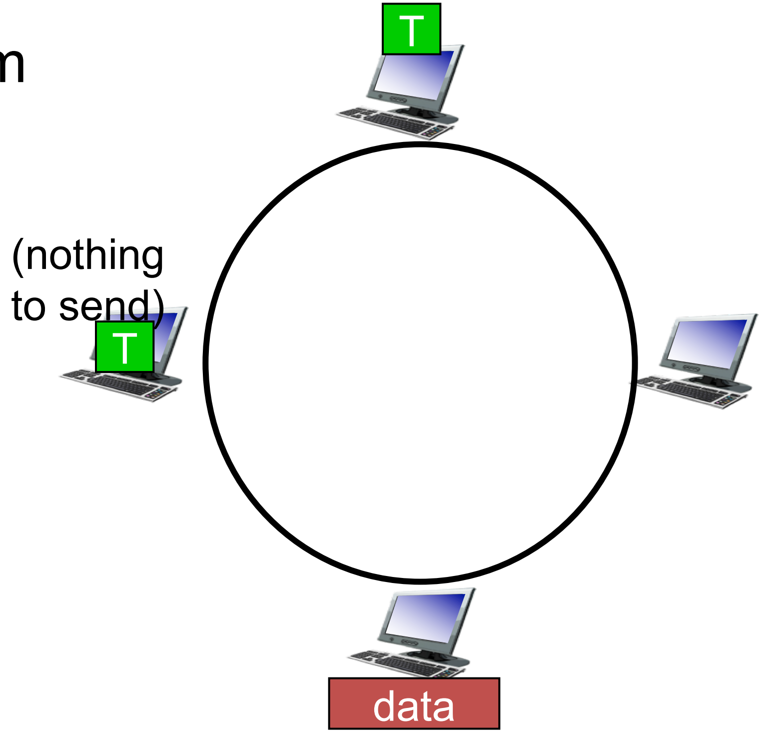
“Taking turns” MAC Protocols

- Polling:
  - master node “invites” slave nodes to transmit in turn
  - concerns:
    - polling overhead
    - single point of failure (master)



“Taking turns” MAC Protocols

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



Summary of MAC Protocols

- Channel partitioning, by time, frequency
  - Time Division, Frequency Division
- Random access (dynamic),
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - CSMA/CD used in Ethernet
  - CSMA/CA used in 802.11
- Taking turns
  - polling from central site, token passing
  - bluetooth

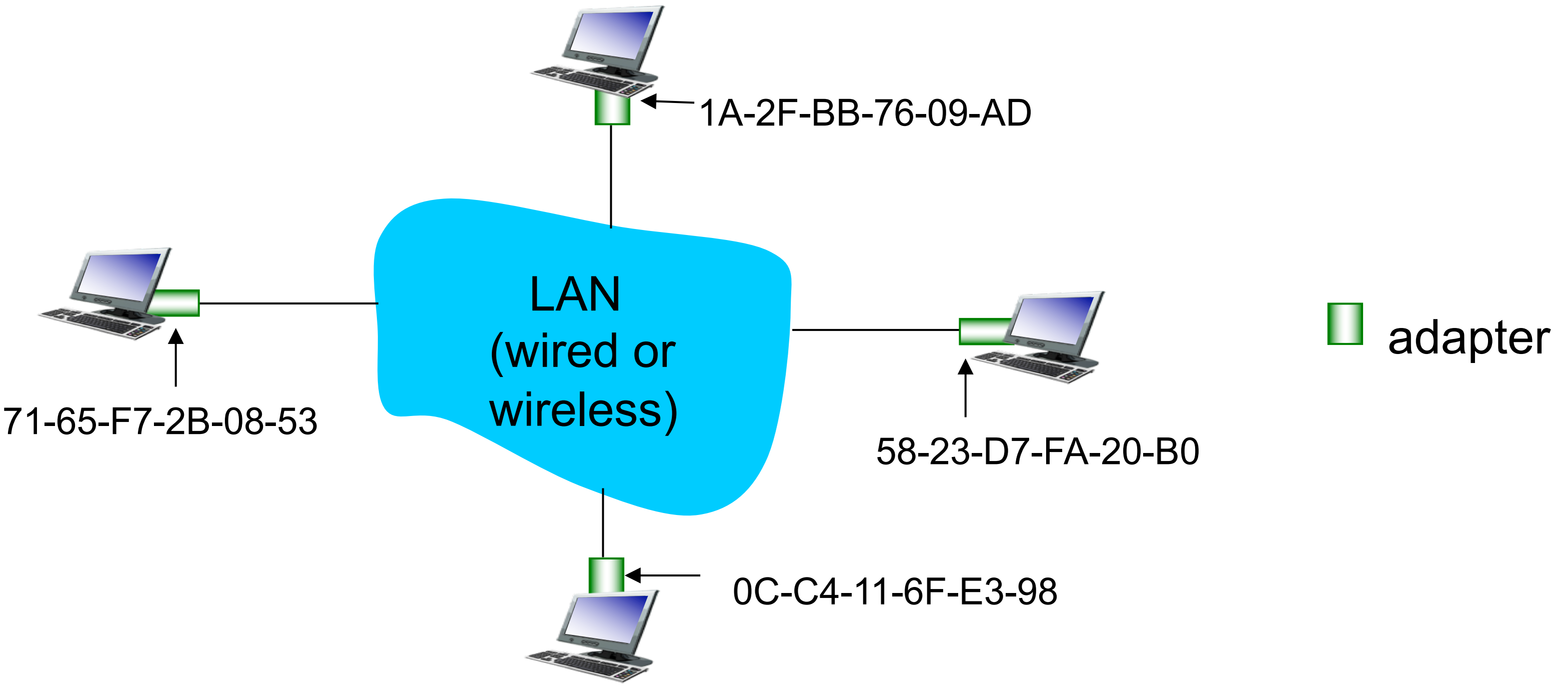
MAC Addresses and ARP

- 32-bit IP address:
  - network-layer address for interface
  - used for layer 3 (network layer) forwarding
- MAC (or LAN or physical or Ethernet) address:
  - function: used 'locally" to get frame from one interface to another physically-connected interface (same network, in IP-addressing sense)
  - 48 bit MAC address (for most LANs) burned in NIC ROM, also sometimes software settable
    - e.g.: 1A-2F-BB-76-09-AD

Hexadecimal notation (each "number" represents 4 bits)

MAC Addresses and ARP

- Each adapter on LAN has unique MAC address

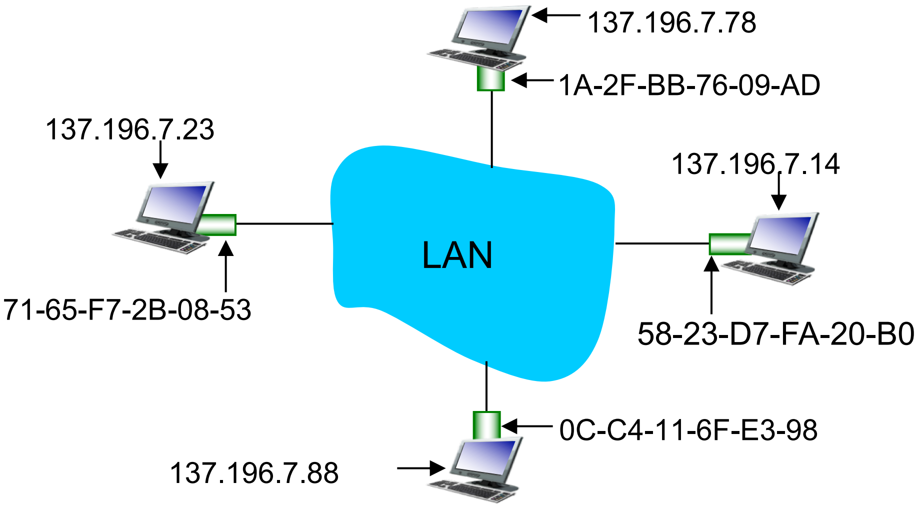


MAC Addresses (More)

- MAC address allocation administered by IEEE
- Manufacturer buys portion of MAC address space (to assure uniqueness)
- Analogy:
  - MAC address: like Social Security Number
  - IP address: like postal address
- MAC flat address → portability
  - can move LAN card from one LAN to another
- IP hierarchical address *not* portable
  - address depends on IP subnet to which node is attached

ARP: Address Resolution Protocol

Question: how to determine interface's MAC address, knowing its IP address?



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

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

ARP protocol: Same LAN

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

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

ARP Request

```
hosta# ping 192.168.1.10
8:0:46:7:4:a3  ff:ff:ff:ff:ff:ff arp 60: arp who-has 192.168.1.10 tell 192.168.1.100
0:1:3:1d:98:b8 8:0:46:7:4:a3     arp 60: arp reply 192.168.1.10 is-at 0:1:3:1d:98:b8
8:0:46:7:4:a3  0:1:3:1d:98:b8    ip  98: 192.168.1.100 > 192.168.1.10: icmp: echo request
0:1:3:1d:98:b8 8:0:46:7:4:a3     ip  98: 192.168.1.10 > 192.168.1.100: icmp: echo reply
hosta# arp -a
hostb (192.168.1.10) at 00:01:03:1D:98:B8 [ether] on eth0
hostb# arp -a
hosta (192.168.1.100) at 08:00:46:07:04:A3 [ether] on eth0
```