Multiple access links protocols

CE 352, Computer Networks

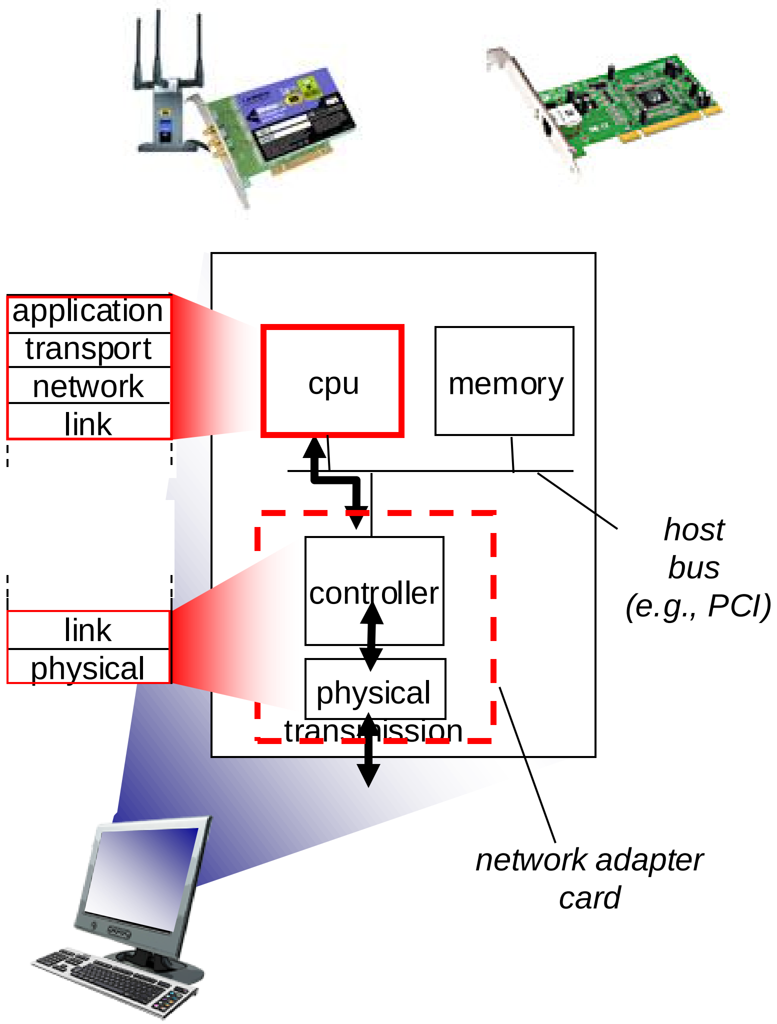
Salem Al-Agtash

Lecture 20

Slides are adapted from Computer Networking: A Top Down Approach, 7th Edition © J.F Kurose and K.W. Ross

Recap (link layer implemented!)

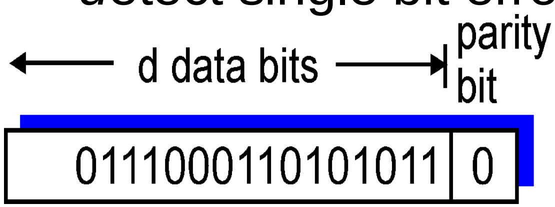
- in each and every host
- link layer implemented in "adaptor" (aka *network interface card* NIC) or on a chip
  - Ethernet card, 802.11 card; Ethernet chipset
  - implements link, physical layer
- attaches into host's system buses
- combination of hardware, software, firmware



Recap (Parity checking)

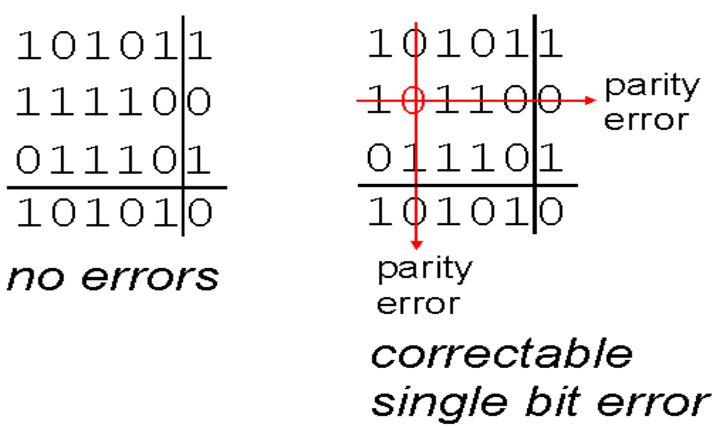
single bit parity:

- *d*etect single bit errors



two-dimensional bit parity:

- detect and correct single bit errors



0

Single bit parity

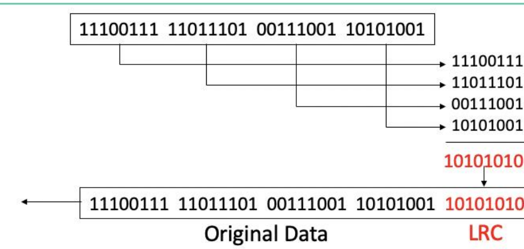
Single bit parity checking is often referred to as vertical redundancy check (VRC).

- Append a single bit at the end of data block such that the number of ones is even (odd)
  → Even Parity (odd parity is similar)
  0110011 → 01100110
  0110001 → 01100011
- Performance:
  - Detects
    - single-bit errors
    - multiple-bit or burst errors only if the total number of errors is odd

COEN 146: Computer Networks Page 14

Longitudinal redundancy check (LRC)

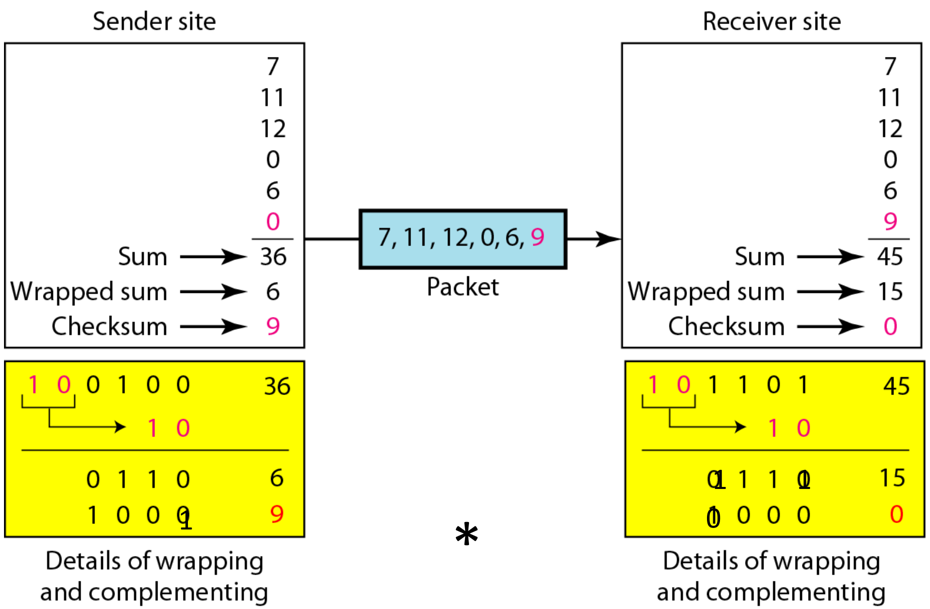
- Organize data into a table and create a parity for each column
- Parity byte
- Detects all single bit and odd bit errors. Some even bit errors are detected



COEN 146: Computer Networks Page 15

Recap (Checksum)

- $1^{'s}$ complement of number 21 using 4-bits → 10101 (5 bits), so wrap the leftmost bit and add it to 4 rightmost bits: 0101+1 = 0110 (6)
- - 6 in 1's complement is 1001 → 9, so complement of 6 is 9
  - Or subtract 6 from $2^n - 1$ (15)
- Example: at the sender: 36 → 100100 in 4-bits = 0100+10 = 0110 (6) by wrapping 2 leftmost bits
  - Checksum is -6
  - in 1's complement→ 9
- at the receiver, 45: 101101
  - Wrapped sum = 1111 (15)
  - In 1's complement -15 = 0000



* Source: Behrouz Forouzan, *Data Communications and Networking*, 4th Edition, McGraw-Hill

Recap (CRC)

want:

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

equivalently:

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

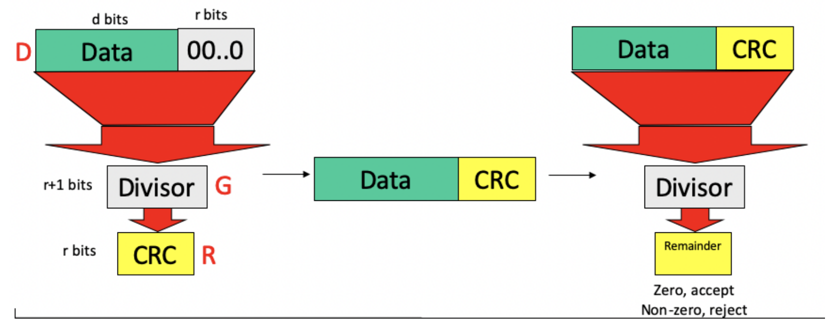
equivalently:

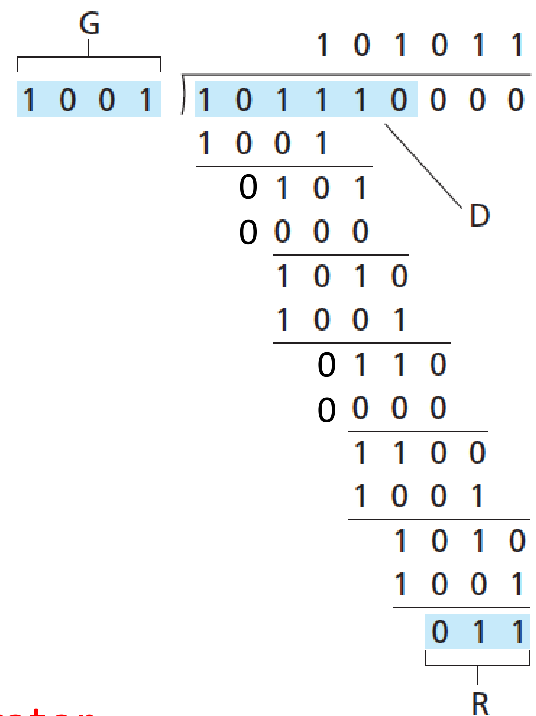
if we divide $D \cdot 2^r$ by G, want remainder R to satisfy:

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

Check the leading left bit is 0 or 1.

- If 0, place a 0 in the quotient and XOR the current bits with 0's.
- If 1, place a 1 in the quotient and XOR the current bits with the divisor



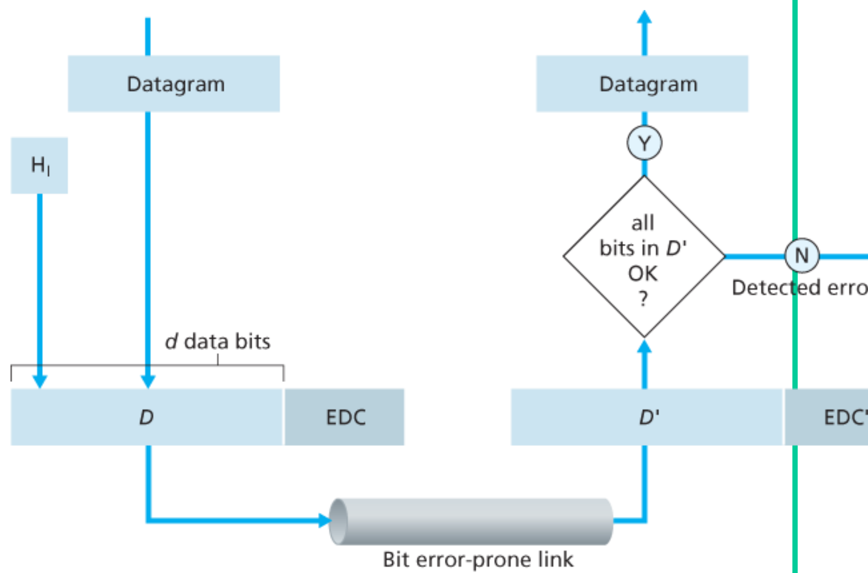


CRC - 8, 12, 16, and 32 bit generator

Example 1

- Suppose the information portion of a packet (D in below diagram) contains 10 bytes consisting of the 8-bit binary representation of the numbers 1 through 10. Compute the internet checksum for this data.

?



Example 1

- Suppose the information portion of a packet (D in below diagram) contains 10 bytes consisting of the 8-bit binary representation of the numbers 1 through 10. Compute the internet checksum for this data.

To compute the Internet checksum (16 bits), we add up (wrapped sum) the values at 16-bit quantities, then we take 1's complement:

00000001 00000010

00000011 00000100

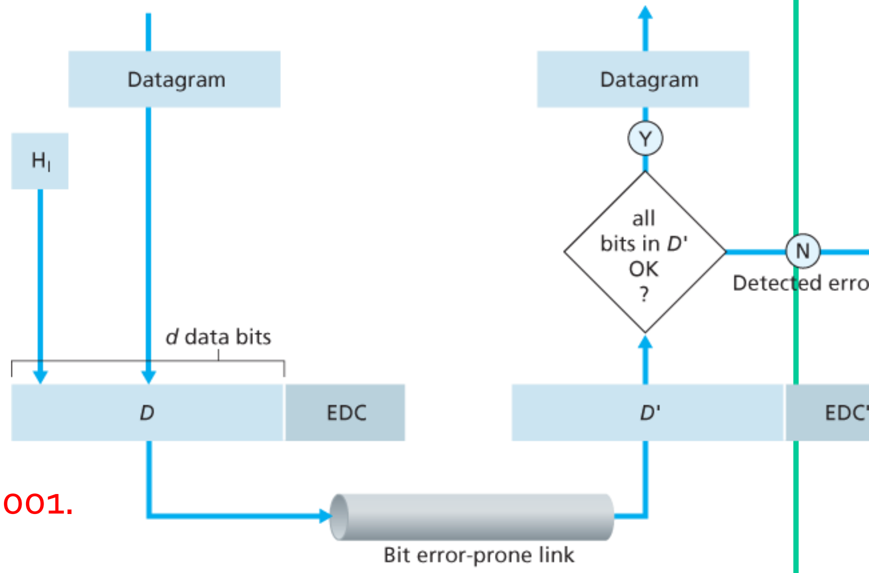
00000101 00000110

00000111 00001000

00001001 00001010

00011001 00011110

The one's complement of the sum is 11100110 11100001.



Example 2

- Consider the 5-bit generator, G=10011, and suppose that D has the value 1010101010. What is the value of R?

?

Example 2

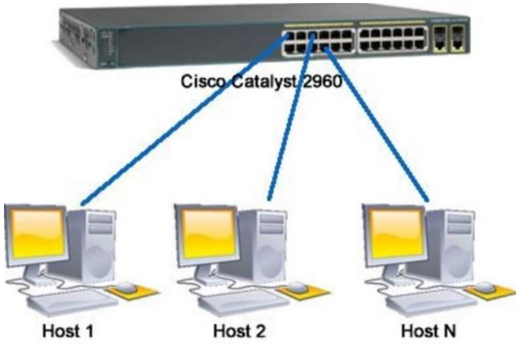
- Consider the 5-bit generator, G=10011, and suppose that D has the value 1010101010. What is the value of R?

```
        1011011100
       ___________
10011 | 1010101010 0000
        10011
         01100
         00000
           11001
           10011
            10100
            10011
             01111
             00000
              11110
              10011
               11010
               10011
                10010
                10011
                      000010
                       00000
                        00100
                        00000
                         0100
```

R=0100

Link types

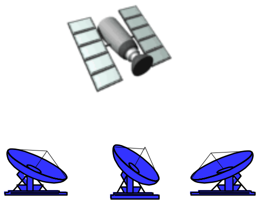
1. ?



2. ?

shared wire (e.g., cabled Ethernet)

shared RF (e.g., 802.11 WiFi)

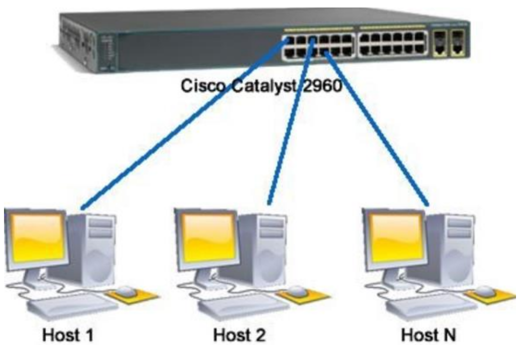


shared RF (satellite)

Link types

1. Point-to-point

- Single sender and single receiver
  - Point-to-point for dial-up access
  - Point-to-point link between Ethernet switch, host
- Protocols
  - Point-to-point protocol (PPP) and High-level data link control (HDLC)

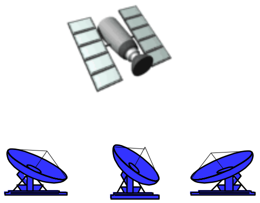


2. Broadcast (shared wire or medium)

- Multiple senders and receivers
  - Ethernet and Wireless LANs

shared wire (e.g., cabled Ethernet)

shared RF (e.g., 802.11 WiFi)



shared RF (satellite)

Multiple access protocols

single shared channel

two or more simultaneous transmissions by nodes → collision

- *collision* if node receives two or more signals at the same time
- frames involved in the collision become tangled together and are lost
- The broadcast channel is wasted during collision intervals

multiple access protocol

distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit

An ideal multiple access protocol

given: broadcast channel of rate *R* bps

Generally:

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 but not easy to implement

MAC protocols:

1. channel partitioning

- divide channel into smaller "pieces"
- allocate piece to node for exclusive use

(1.1 time slots, 1.2 frequency, 1.3 code)

2. random access

- channel not divided, allow collisions
- "recover" from collisions

(2.1 Slotted ALOHA, 2.2 Pure ALOHA, 2.3 CSMA, CSMA/CD, CSMA/CA)

3. "taking turns"

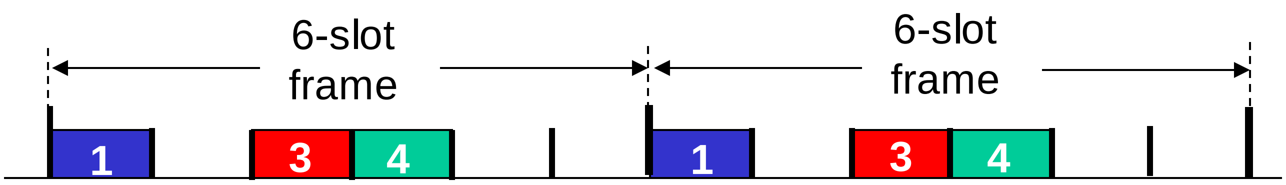
- nodes take turns, but nodes with more to send can take longer turns

(3.1 Polling, 3.2 token passing)

1.1 TDMA

Channel partitioning MAC protocol - time division multiple access

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

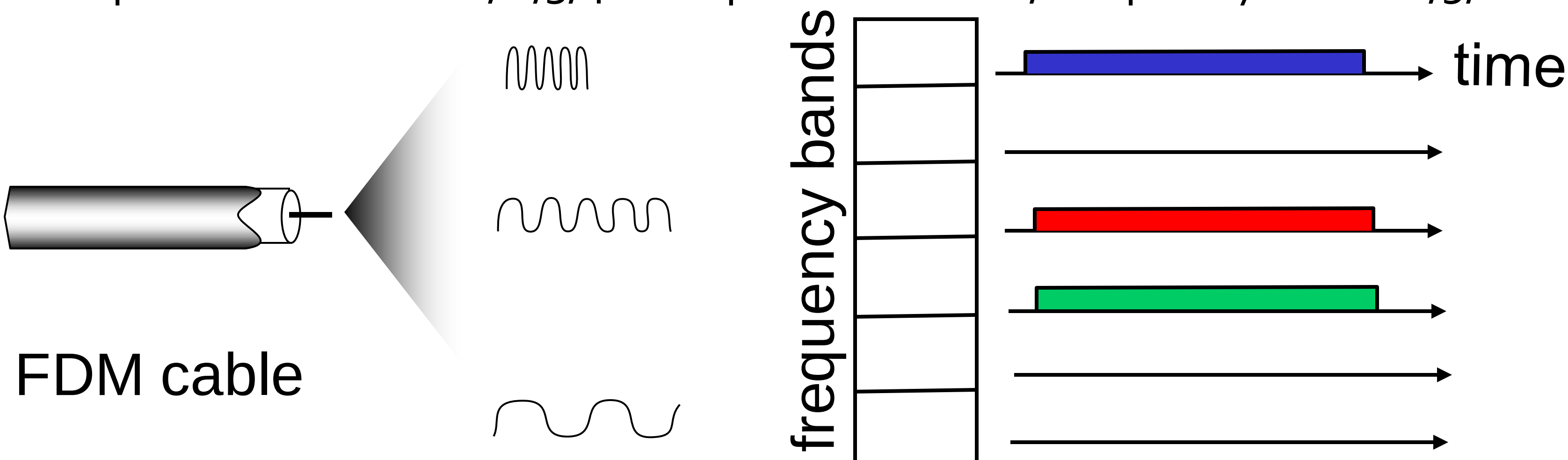


- N nodes, Rbps link → each nodes transmission rate = R/N bps
- Pros: no collision and fair
- Cons: R/N bps at maximum and must wait N-1 time slots, even if no other node is transmitting

1.2 FDMA

Channel partitioning MAC protocol: 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 packet to send, frequency bands 2,5,6 idle



- N nodes, Rbps link → each nodes transmission rate = R/N bps
- Pros: no collision and fair
- Cons: R/N bps at maximum even if no other node is transmitting

1.3 CDMA

Channel partitioning MAC protocol: code division multiple access

- Code is used instead of time or frequency slots
- entire bandwidth is being used by users all the time
- each station encodes its data using the assigned code
- Widely used in wireless communication
  - CDMA (3G, 4G, 5G)
- Details to follow in Ch7

2. 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:

2.1 slotted ALOHA

2.2 ALOHA

2.3 CSMA, CSMA/CD, CSMA/CA

2.1 Slotted ALOHA

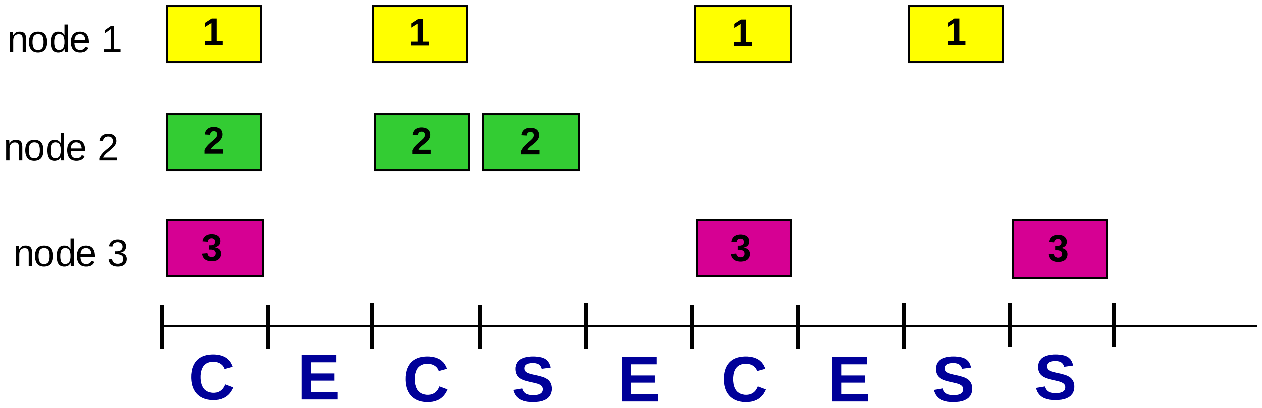
assumptions:

- all frames same size – L bits
- time divided into equal size slots (time to transmit 1 frame) – L/R
- nodes start to transmit only slot beginning
- nodes are synchronized with slot timing
- if 2 or more nodes transmit in slot, all nodes detect collision before slot time ends

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 *probability p* until success



C: Collision slot (waste)
E: Empty slot (no use)
S: Successful slot

Efficiency

- *Efficiency* is defined as the long-run fraction of successful slots in the case when there are a large number of active nodes, each always having a large number of frames to send.
- If nodes immediately send when collision occurs, efficiency → 0
- *N* nodes with many frames therefore need to coordinate to send,
  - each transmits in slot with probability *p*
  - each skips to transmit in slot with probability *(1-p)*
- P(success by GIVEN node in a slot) =

  P(a node transmits and remaining N-1 nodes do not transmit) $= p(1-p)^{N-1}$
- P(success by ANY node in a slot) = $= Np(1-p)^{N-1}$
- Max efficiency: find *p** that maximizes

  $$E(p) = Np(1-p)^{N-1}$$

Max efficiency

$$E'(p) = N(1-p)^{N-1} - Np(N-1)(1-p)^{N-2} \longrightarrow = N(1-p)^{N-2}((1-p) - p(N-1)) \longrightarrow E'(p) = 0 \Rightarrow p^* = \frac{1}{N}$$

- for many nodes, take limit of $Np^*(1-p^*)^{N-1}$ as N goes to infinity, gives:

$$E(p^*) = N\frac{1}{N}(1-\frac{1}{N})^{N-1} = (1-\frac{1}{N})^{N-1} = \frac{(1-\frac{1}{N})^N}{1-\frac{1}{N}} \longrightarrow \lim_{N\to\infty}(1-\frac{1}{N})^N = \frac{1}{e}$$

- *max efficiency = 1/e = .37*

  → at best: channel useful transmissions 37% of time!

- E.g. On a 100-Mbps slotted ALOHA system, a successful throughput on this channel will be less then 37%

2.1 Slotted ALOHA

Pros:

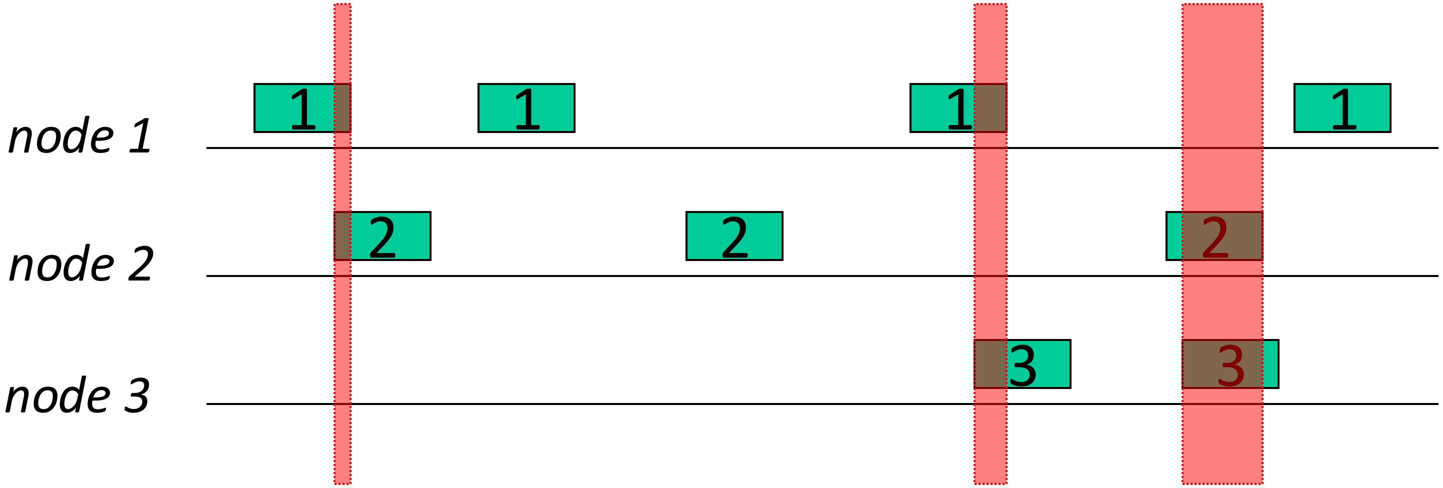
- single active node can continuously transmit at full rate of channel
- highly decentralized: only slots in nodes need to be in sync
- simple

Cons:

- collisions, wasting slots
- idle slots, no use
- nodes may be able to detect collision in less than time slot to transmit packet
- clock synchronization, because transmission is only during slots

2.2 Pure (unslotted) ALOHA

- Pure Aloha: simpler, no synchronization, when a frame first arrives
  - transmit immediately
- If collision is detected, the node will then retransmit with probability p
- Else, the node waits, then not retransmit with probability $1 - p$
- Collision probability increases:
  - frame sent may collide with other frames sent



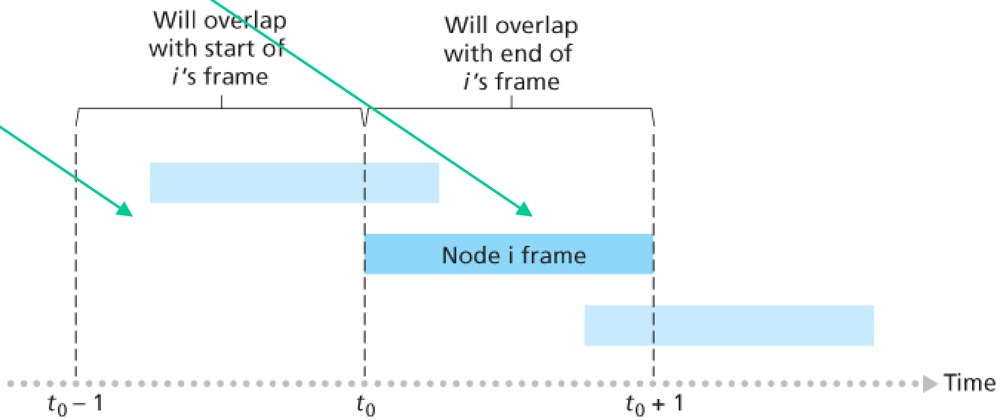
Efficiency

P(success by given node) =

P(node transmits) · P(no other node transmits in a given interval) [Two intervals]

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

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



… choosing optimum p and then letting *n goes to infinity*

$= 1/(2e) = .18$

even *worse* than slotted Aloha!

2.3 CSMA (carrier sense multiple access)

CSMA: listen before transmit:

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

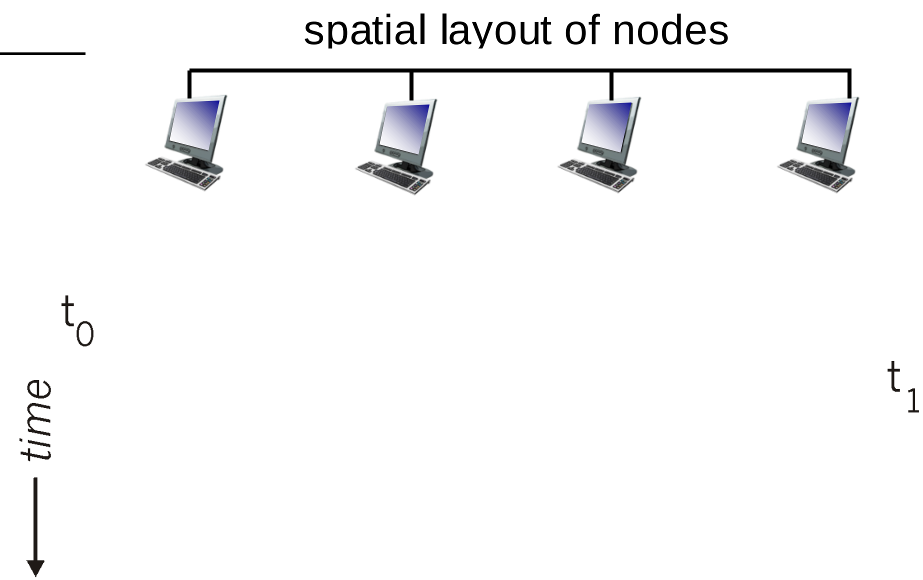
In ALOHA, a node's decision to transmit is made independently

Human conversation (polite) analogy:

- **Listen before speaking.** If someone else is speaking, wait until they are finished. In the networking world, this is called **carrier sensing**—a node listens to the channel before transmitting. If a frame from another node is currently being transmitted into the channel, a node then waits until it detects no transmissions for a short amount of time and then begins transmission.
- **If someone else begins talking at the same time, stop talking.** In the networking world, this is called **collision detection**—a transmitting node listens to the channel while it is transmitting. If it detects that another node is transmitting an interfering frame, it stops transmitting and waits a random amount of time before repeating the sense-and-transmit-when-idle cycle.

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 determining collision probability



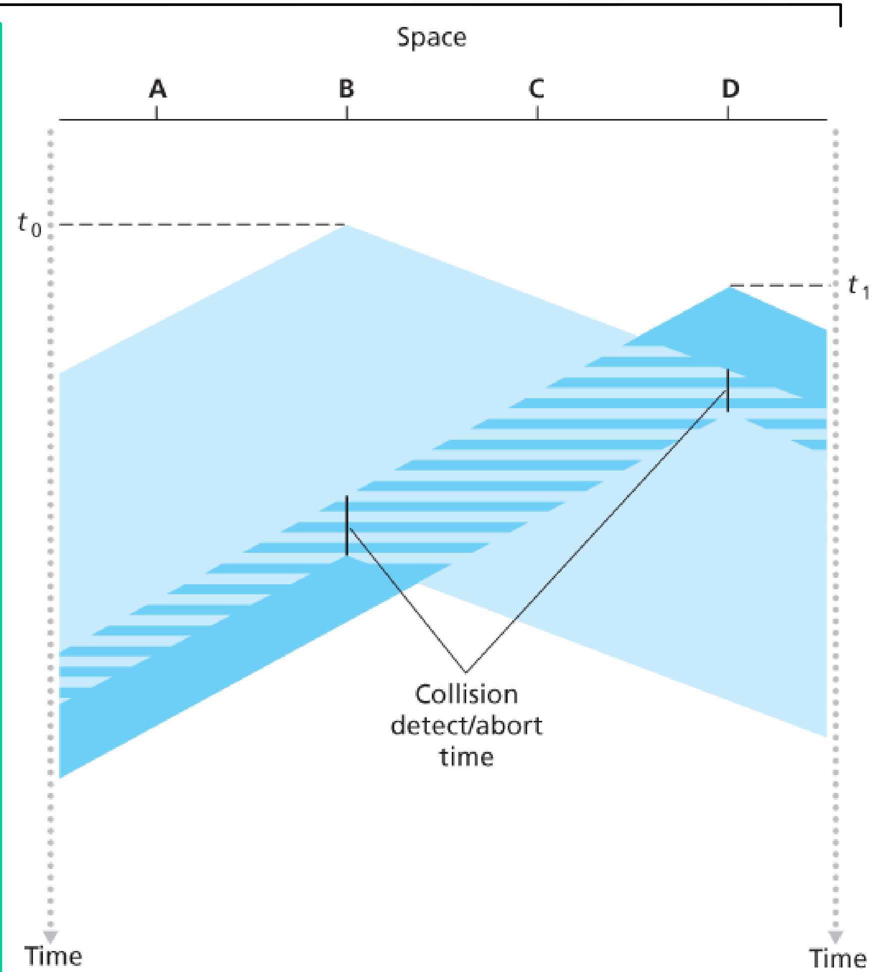
CSMA/CD (collision detection)

CSMA/CD: carrier sensing

- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage
- waits a random amount of time, then retransmits

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



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 $\{0,1,2, \ldots, 2^m-1\}$.
   - In Ethernet: NIC waits K·512 bit times, then go to Step 2
   - longer backoff interval with more collisions , In Ethernet: m is capped at 10

e.g. after the second collision, k is chosen with equal probability from {0,1,2,3}, after 10 collisions or more, k is chosen with equal probability from {0,1…1023}

- For 100 Mbps Ethernet, NIC waits for k.5.12 microseconds

Efficiency

Efficiency (better than ALOHA):

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

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

- as t_{prop} approaches 0, the efficiency approaches 1, meaning colliding nodes will abort immediately without wasting the channel
- as t_{trans} becomes very large, efficiency approaches 1, meaning when a frame grabs the channel, it will hold on to the channel for a very long time; thus, the channel will be doing productive work most of the time

3. Taking turns

3.1 polling (Bluetooth):

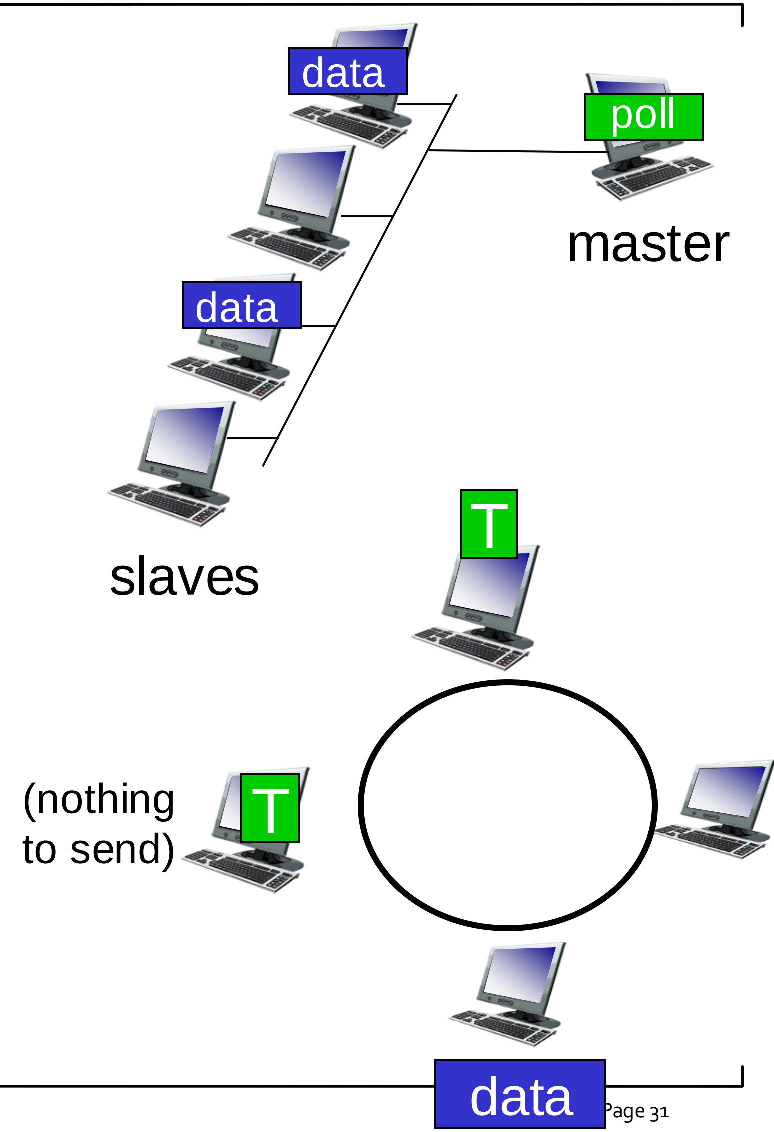
Master node "invites" slave nodes to transmit in turn → concerns:

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

3.2 token passing (fiber distributed data interface (FDDI):

control *token* passed from one node to next sequentially → concerns:

- token overhead
- Latency
- single point of failure (token)



Summary

Today:

- Multiple access protocols
  - Partitioning (TDMA, FDMA, CDMA)
  - random access (ALOHA, CSMA)
  - taking turns (polling, token passing)

Canvas discussion:

- Reflection
- Exit ticket

Next time:

- read 6.4 of KR (LAN)
- follow on Canvas! material and announcements

Any questions?