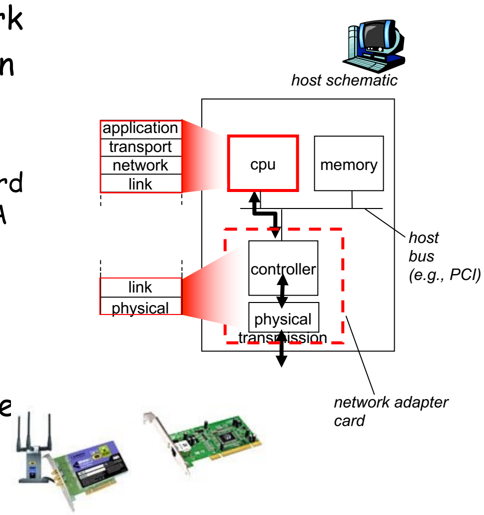
Lecture 25

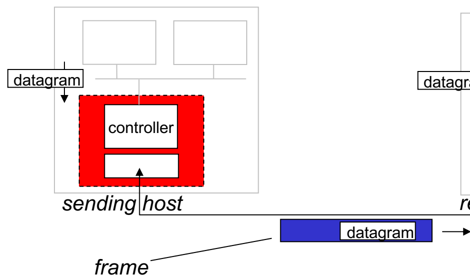
- Sections 6.1.2, 6.2 (6.2.1, 6.2.2 and 6.2.3) and 6.3 (6.3.1 and 6.3.2)
  - Where is the link layer implemented
  - Error detection and correction techniques
    - Parity
    - Checksum method
    - Cyclic redundancy check
  - Multiple access protocols
    - Channel partitioning protocols
    - Random access protocols

Where is the link layer implemented?

- in each and every host that connects to network
- link layer implemented in "adaptor" (aka *network interface card* NIC)
  - Ethernet card, 802.11 card (built in, USB, or PCMCIA card) 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, rdt, flow control, etc.
  - uses an access protocol to send protocol on medium
- receiving side
  - looks for errors, rdt, flow control, etc
  - extracts datagram, passes to upper layer at receiving side

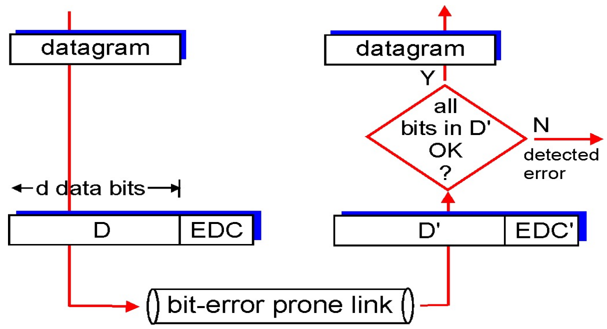
Link Layer

6.1 Introduction and services

6.2 Error detection and correction

Error Detection

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



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

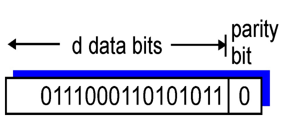
Data Link Layer 6-14

Parity Checking

Single Bit Parity:

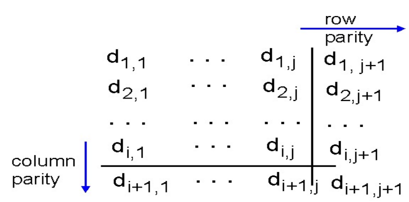
Parity bit = XNOR of all data bits (odd parity) – even parity can be used

Detect single bit errors

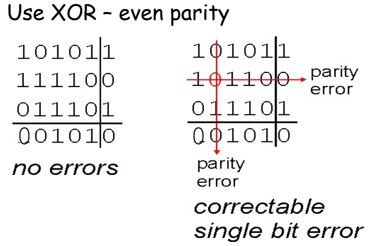


Two Dimensional Bit Parity:

Detect *& correct* single bit errors



Use XOR – even parity



Data Link Layer 6-15

Internet checksum (review)

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

Sender:

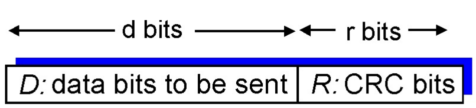
- treat segment contents as sequence of 16-bit integers
- checksum: addition (1's complement sum) of segment contents
- sender puts checksum value into segment checksum field

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 a binary number
- choose r+1 bit pattern (generator), *G*
- goal: compute 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 Rules

- G: r+1 bits
  - LSB and MSB must be 1
- Sender: divides $D.2^r$ by G ($D.2^r = nG \oplus R$)
  - Quotient bit is 1 if MSB in partial dividend is 1; otherwise 0
  - Addition and subtraction is bit-wise XOR (no carry and no borrow)
  - Remainder, R, is r bits long (*ignore quotient, n)*
  - Sends $D.2^r \oplus R$ (bit-wise XOR) (which is nG)
- Receiver: divides received data by G
  - 0: no error; otherwise, error

Data Link Layer 6-18

18

CRC Example

Want:

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

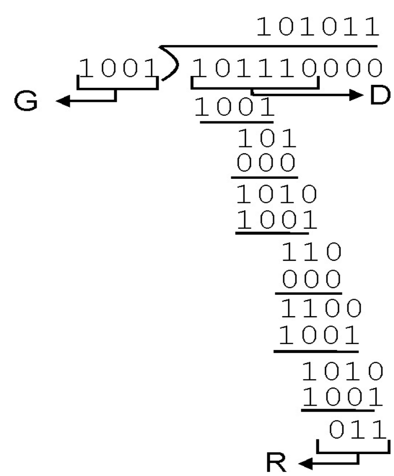
equivalently:

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

equivalently:

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

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



Data Link Layer 6-19

19

Error & No Error

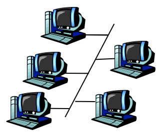
Link Layer

6.1 Introduction and services
6.2 Error detection and correction
6.3Multiple access protocols

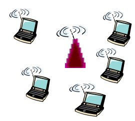
Multiple Access Links and Protocols

Two types of "links":

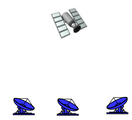
- point-to-point
  - PPP for dial-up access
  - point-to-point link between Ethernet switch and host
- broadcast (shared wire or medium)
  - old-fashioned Ethernet; Ethernet hub
  - upstream HFC
  - 802.11 wireless LAN



shared wire (e.g., cabled Ethernet)



shared RF (e.g., 802.11 WiFi)



shared RF (satellite)

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

- distributed algorithm that 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

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

Does not exist in practice

MAC Protocols: a taxonomy

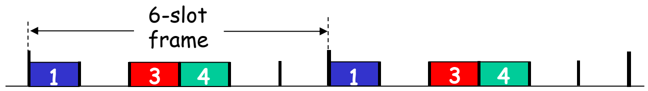
Three broad classes:

- 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 (e.g., by passing tokens), 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 = pkt trans time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have pkt, slots 2,5,6 idle



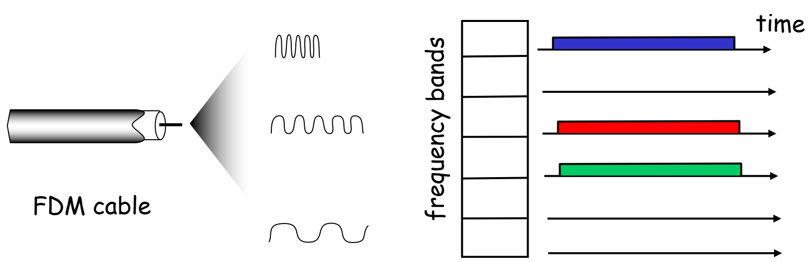
Data Link Layer 6-26

26

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



Data Link Layer 6-27

27

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:
  - when to transmit (or try to avoid collisions)
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed and randomized retransmissions)
- Examples of random access MAC protocols:
  - slotted ALOHA
  - (pure) ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA

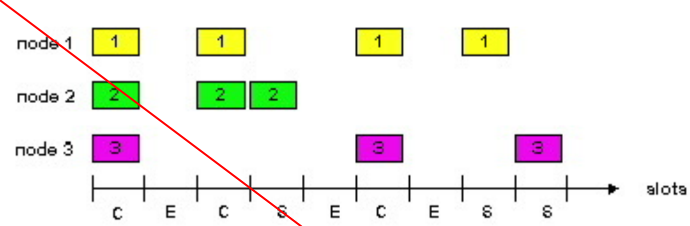
Assumptions:

- 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

Data Link Layer 5-30

30

Slotted Aloha efficiency

Efficiency : long-run fraction of successful slots (many nodes, all with many frames to send)

- *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 = .37

At best: channel used for useful transmissions 37% of time!

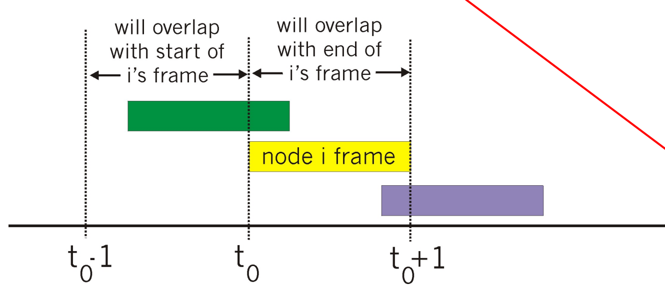
!

Data Link Layer 5-31

31

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 $[t_0-1, t_0+1]$



Data Link Layer 5-32

32

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* than slotted Aloha!

Data Link Layer 5-33

33

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!

Can collisions still occur?

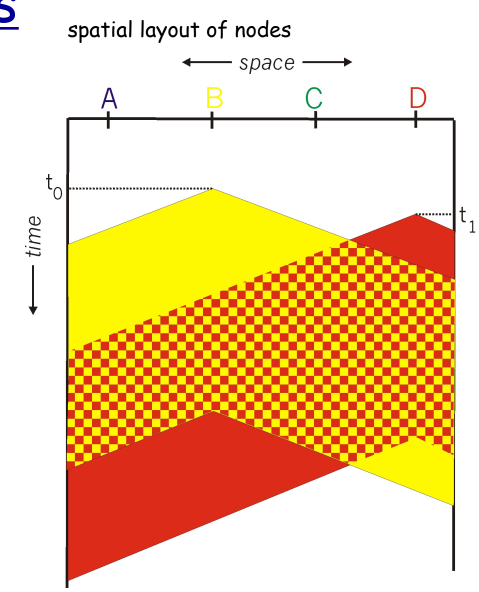
CSMA collisions

collisions *can* still occur:

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

collision:

entire packet transmission time wasted

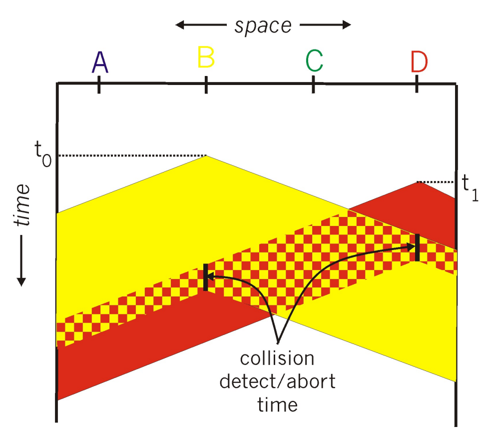


CSMA/CD (Collision Detection)

CSMA/CD: carrier sensing, deferral as in CSMA

- listen after transmission for collisions:
  - if there is a collision:
    - collision *detected* within short time, and stop transmission (colliding transmissions aborted), hence reducing channel wastage
    - reschedule collided transmission after some random time
- 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



What is the maximum time to detect a collision, if any?