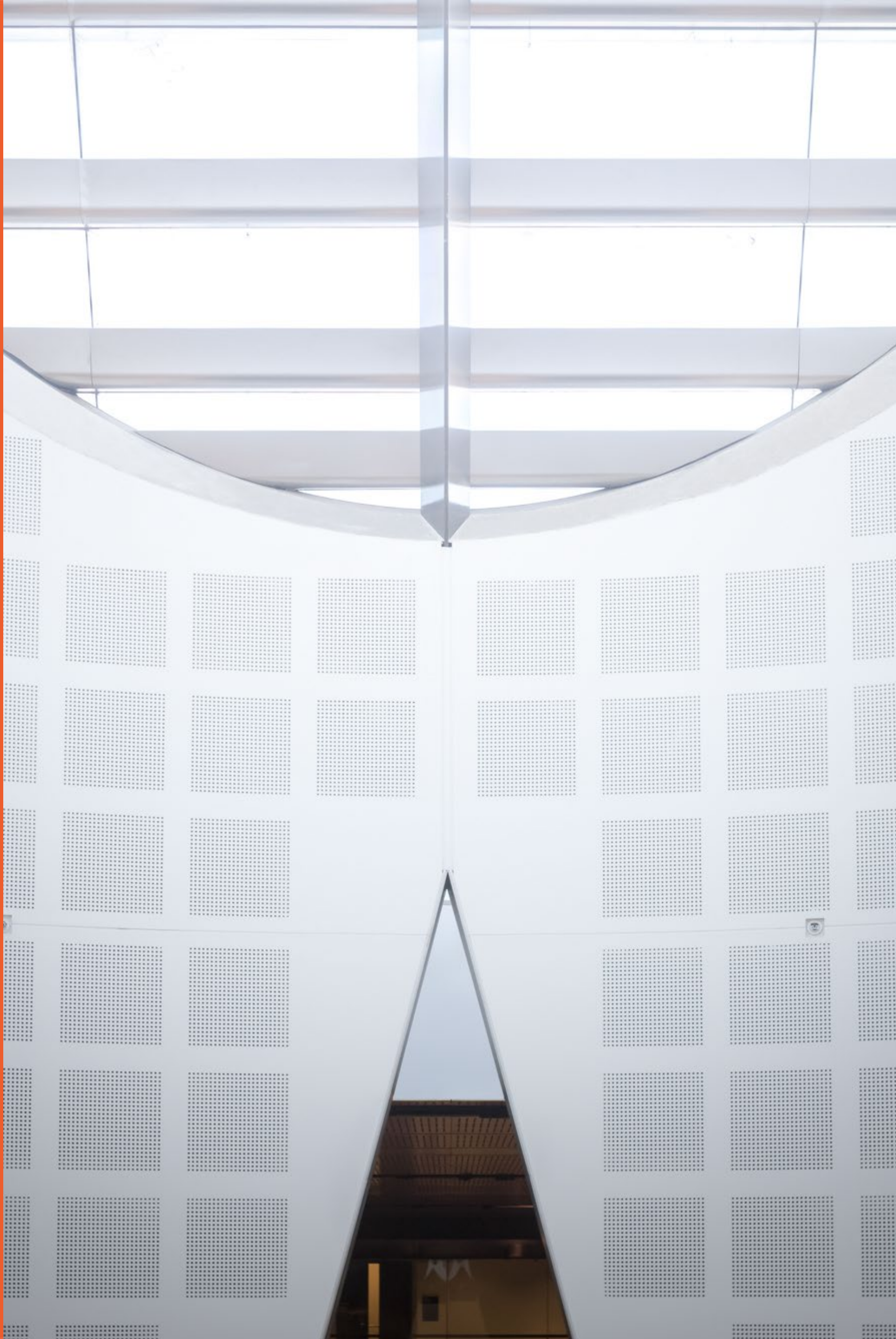
COMP9121: Design of Networks and Distributed Systems

Week 2: Link Layer 1

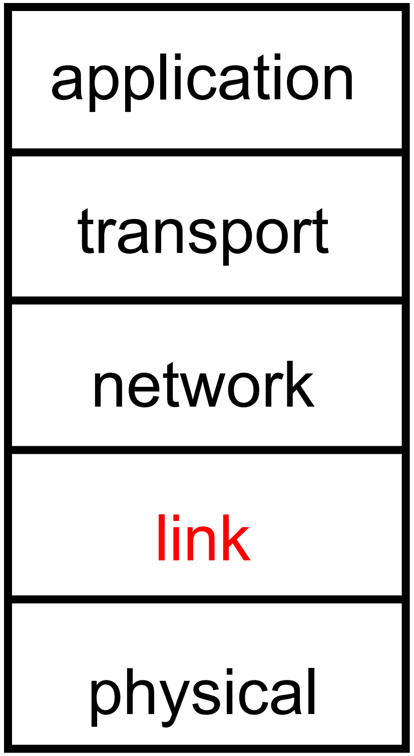
Wei Bao

School of Computer Science

THE UNIVERSITY OF SYDNEY



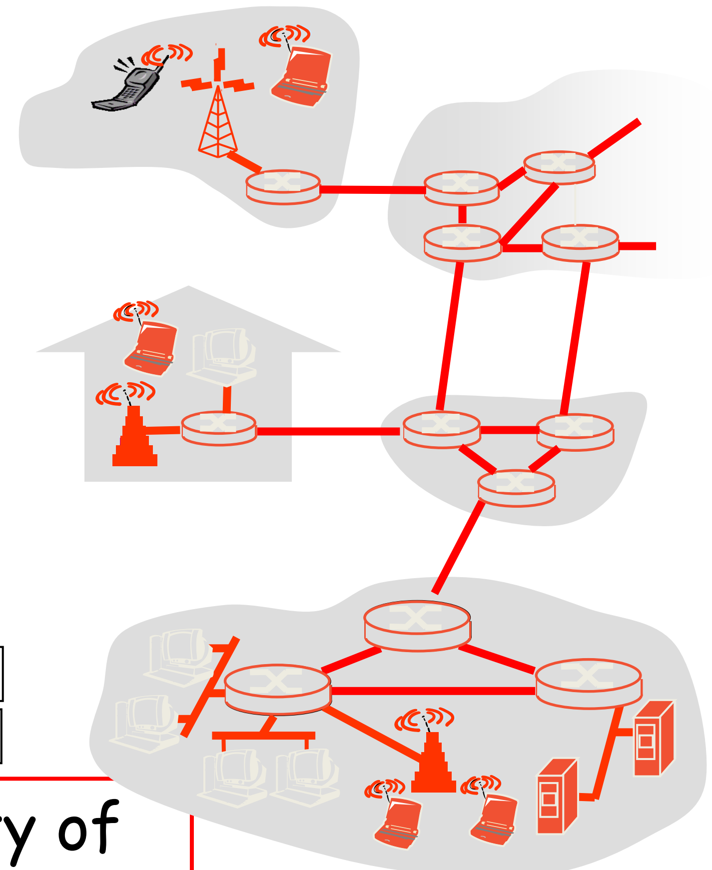
Last week



Link Layer

- Introduction and services
- Error detection and correction
- Multiple access protocols
- Link-layer Addressing

Link Layer: Introduction



Some terminology:

- hosts and routers are **nodes**
- communication channels that connect adjacent nodes along communication path are **links**
  - wired links
    - LANs
  - wireless links
    - WLAN
- layer-2 packet is a **frame**, encapsulates (network layer) datagram

datagram		H_n	H_t	M
frame	H_l	H_n	H_t	M

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

Link Layer Services

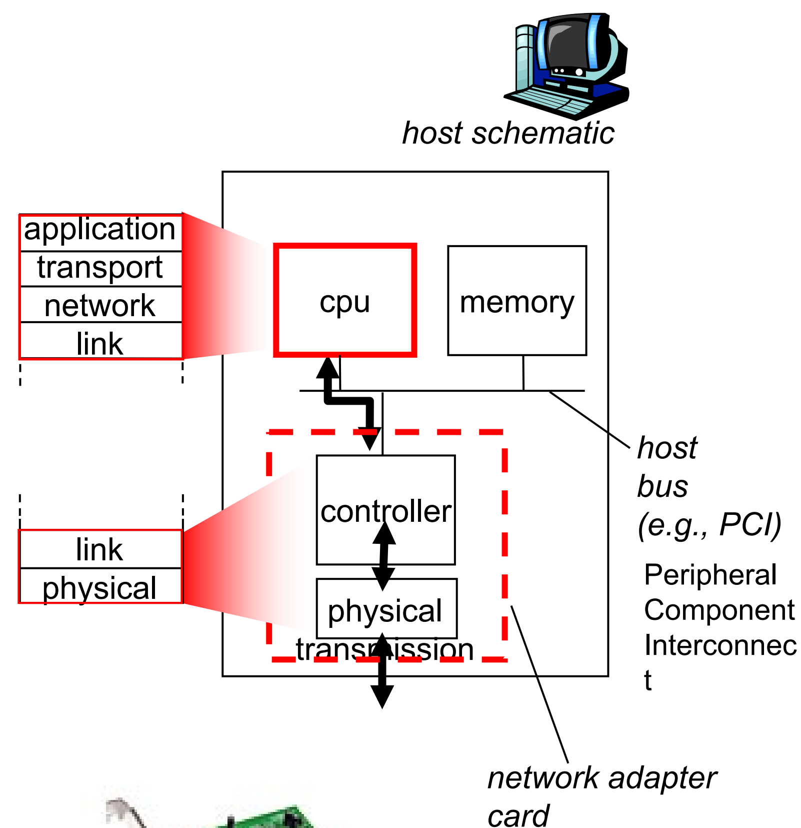
- *framing, link access:*
  - encapsulate datagram into frame, adding header
  - channel access if shared medium
  - "MAC" addresses used in frame headers to identify source, destination over one link
    - different from IP address!

Link Layer Services (more)

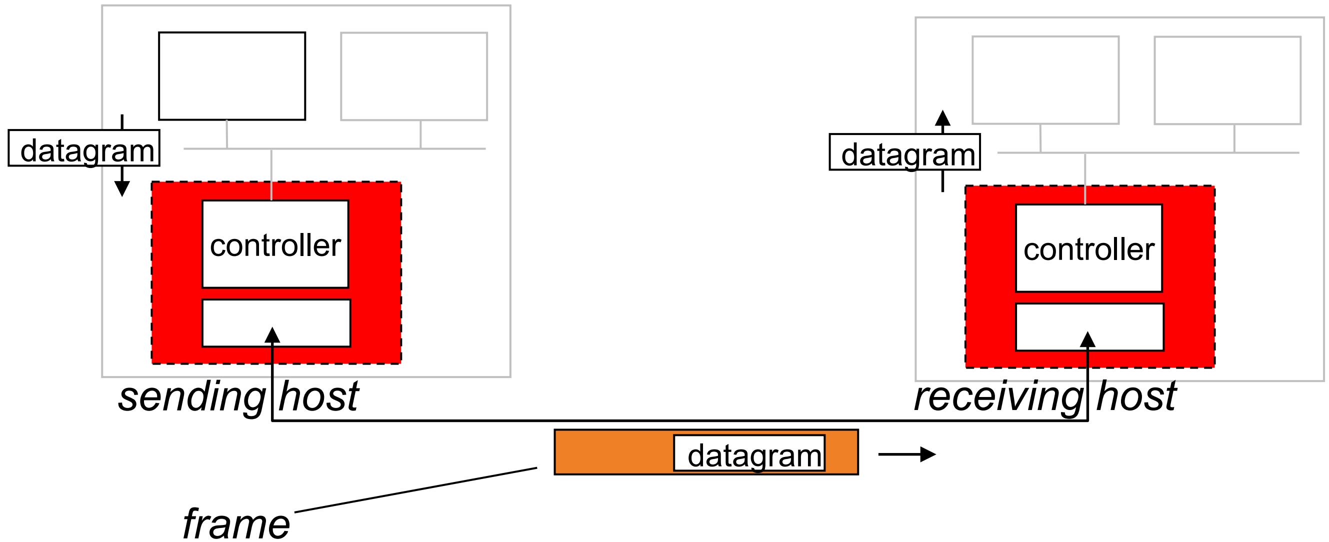
- *error detection*:
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
- error correction:
  - receiver identifies *and corrects* bit error(s) without resorting to retransmission

Where is the link layer implemented?

- in each and every host
- link layer implemented in "adaptor" (aka *network interface card* NIC)
  - implements link and physical layers
- attaches into host's system buses
- combination of hardware, software, firmware



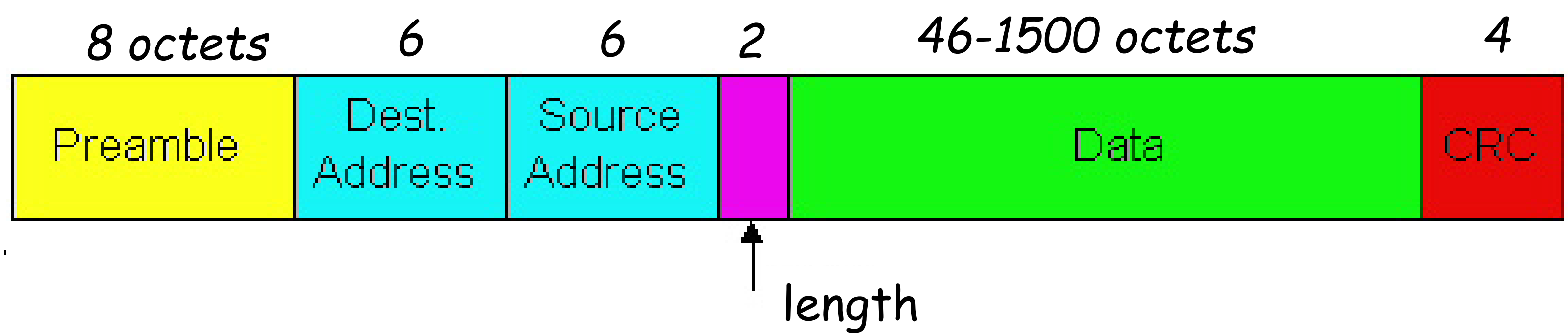
Adaptors Communicating



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

Example of Frame Structure

– IEEE 802.3 frame structure
  – CRC bits are used for error detection and are appended as a trailer.



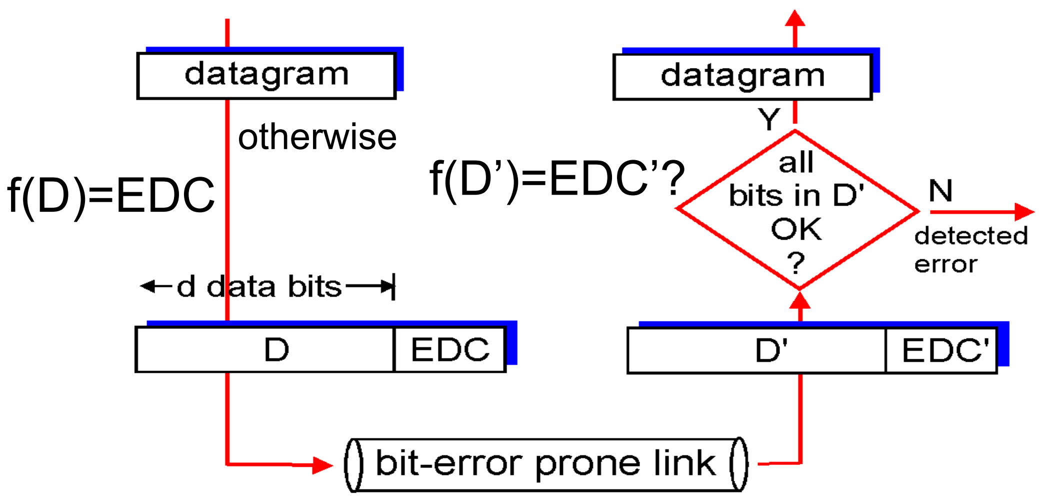
Link Layer

- Introduction and services
- Error detection and correction
- Multiple access protocols
- Link-layer Addressing

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

d data bits parity bit
110100 1

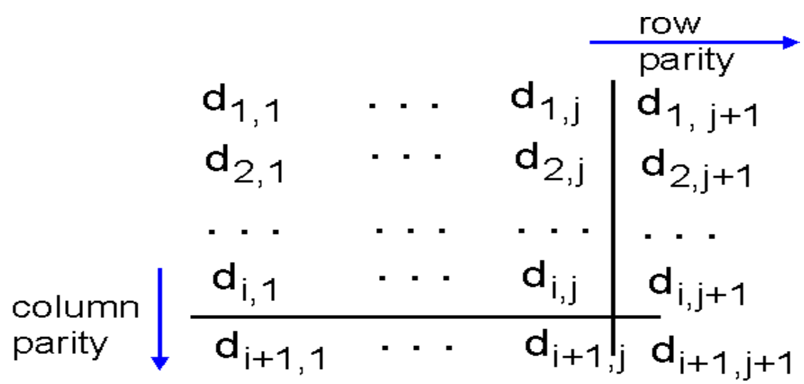
odd number of ‘1’s -> 1

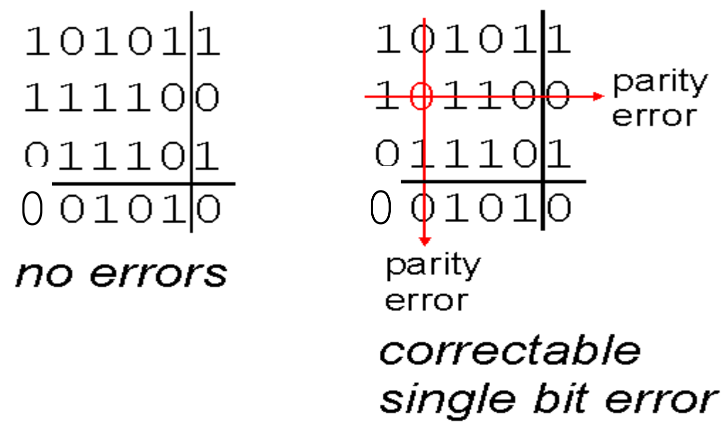
even number of ‘1’s -> 0

total number of ‘1’s -> even

Two Dimensional Bit Parity:

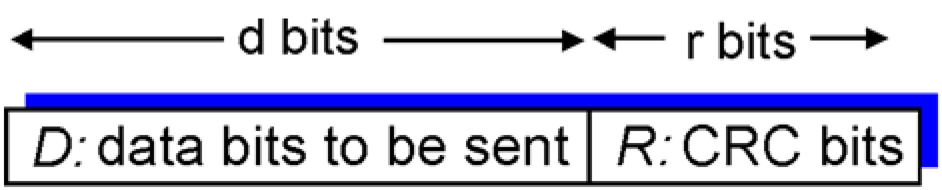
Detect *and correct* single bit errors





Cyclic Redundancy Check (CRC)

- d data bits
- r CRC bits
- Sender: send d+r bits
- Receiver: check d+r bits
- Q: How to design the r bits?



Design Intuition

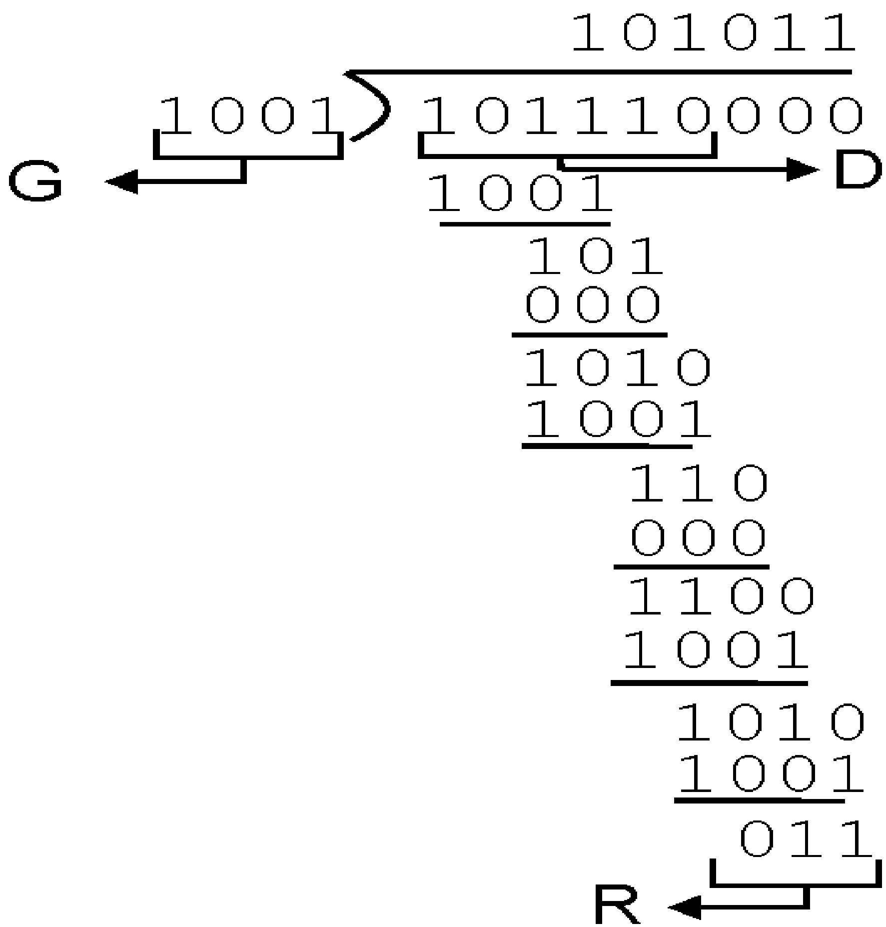
- Consider decimal
- Want to send 12345
- Consider to send 12345+redundancy
- Choose a divisor, e.g., 21 (known by sender and receiver)
- Consider 12345*100/21
  - Not divisible!
  - Quotient 58785
  - Remainder 15
- Send 1234500+(21-15)=1234506
  - 12345 information
  - 06 redundancy
- 1234506/21 divisible!
- Receiver knows 21,
  - If receives 1234406: not divisible by 21. Error detect!

Cyclic Redundancy Check (CRC)

- Binary, modulo 2 domain
- addition, subtraction
- 1+1=0, 1-1=0,
- 1+0=1, 1-0=1
- 0+1=1, 0-1=1
- 0+0=0, 0-0=0
- “-”, “+”, are equivalent to, XOR, ⊕
- 11+11=00: no carry over
- Multiplication
- 11*100=11*2^2=1100 (left shift 2 bits)
- 11*11=11*10+11*1=110+11=101

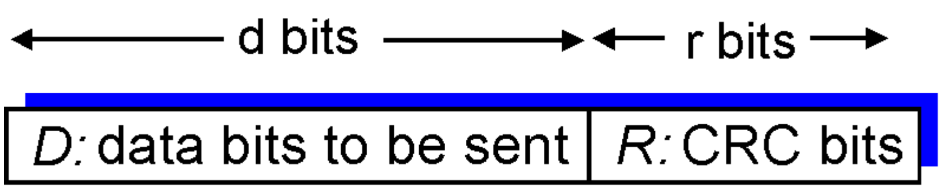
Cyclic Redundancy Check (CRC)

- Division



Cyclic Redundancy Check (CRC)

- 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

$$<D,R> = D * 2^r \text{ XOR } R$$

mathematical formula

CRC Example

Goal:

$D \cdot 2^r + R$ divisible by G

Approach:

Divide $D \cdot 2^r$ by G, get remainder R

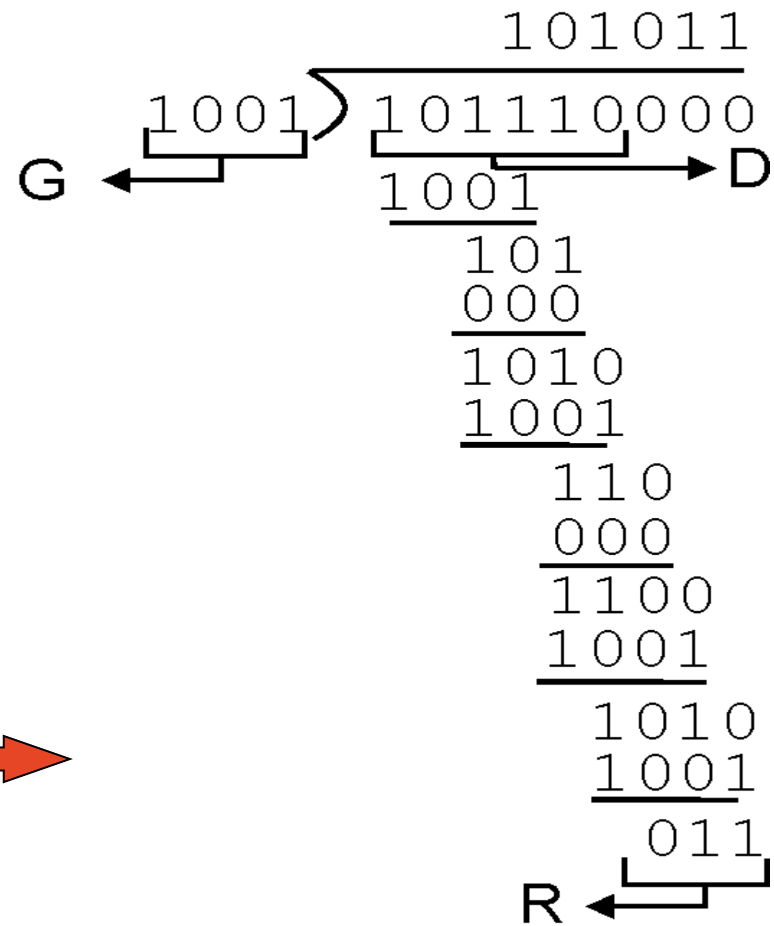
$$R = \text{remainder}\left[\frac{D \cdot 2^r}{G}\right]$$

$D \cdot 2^r + R = D \cdot 2^r - R = D \cdot 2^r \oplus R$,
divisible by G

$D \cdot 2^r \oplus R$ is what we want

Example : D = 101110, G=1001

CRC coded: 101110<u>011</u>



Note: remainder[101110011/1001]=0

CRC Example

	Implementation	Math Explanation
Begin	D=101110 G=1001	Given D, G
	Generator r+1 bits Left shift r bits	
Step 1	Left shift 101110 3 bits 101110000	Generate $D \cdot 2^r$
Step 2	Remainder[101110000/1001]=011	R= remainder [$D \cdot 2^r$ /G]
Step 3	To send: 101110 011	$D \cdot 2^r + R$
	Remainder [101110 011/1001]=0	$D \cdot 2^r + R$ is divisible by G
Step 5	Receiver: check if the received bits are divisible by 1001	$D \cdot 2^r + R$ is divisible by G there is no error

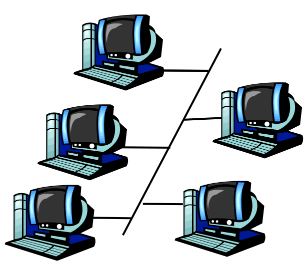
Link Layer

- Introduction and services
- Error detection and correction
- Multiple access protocols
- Link-layer Addressing

Multiple Access Links and Protocols

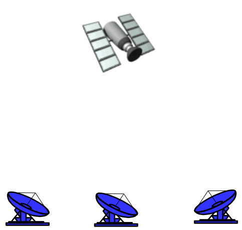
Two types of "links":

- point-to-point
- broadcast (shared wire or medium)



shared wire (e.g., cabled Ethernet)

shared RF (e.g., 802.11 WiFi)



shared RF (satellite)

humans at a cocktail party (shared air, acoustical)

Multiple Access protocols

- single shared broadcast channel
- two or more simultaneous transmissions by nodes
  - 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

<u>Broadcast channel of rate R bps</u>

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

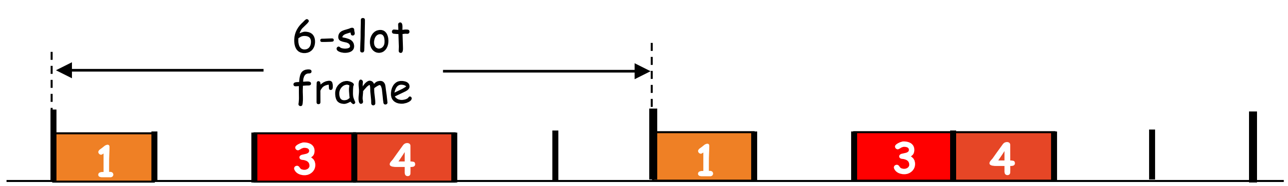
MAC Protocols: a taxonomy

Three broad classes:

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

Channel Partitioning MAC protocols: TDMA

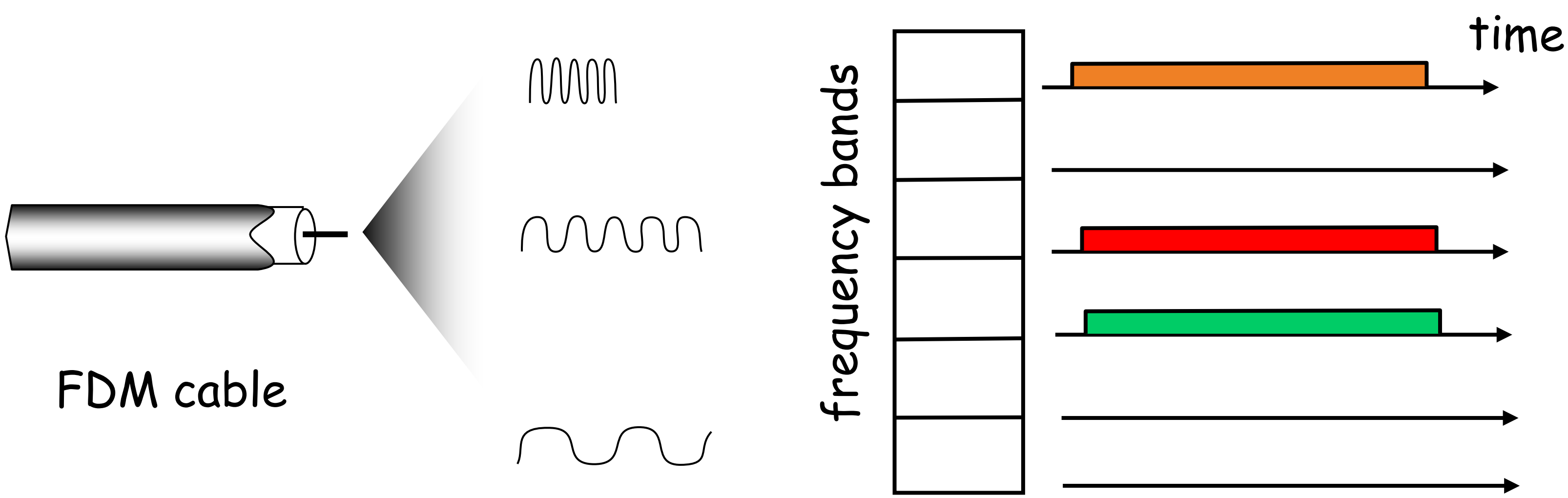
- TDMA: time division multiple access
- access to channel in "rounds"
- each station gets fixed length slot (length = pkt trans time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have pkt, slots 2,5,6 idle



Channel Partitioning MAC protocols: FDMA

FDMA: frequency division multiple access

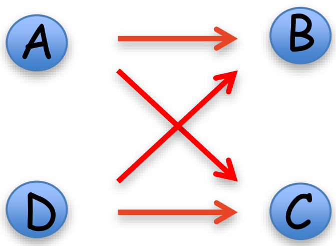
- channel spectrum divided into frequency bands
- each station assigned fixed frequency band
- unused transmission time in frequency bands go idle
- example: 6-station LAN, 1,3,4 have pkt, frequency bands 2,5,6 idle



Random Access Protocols

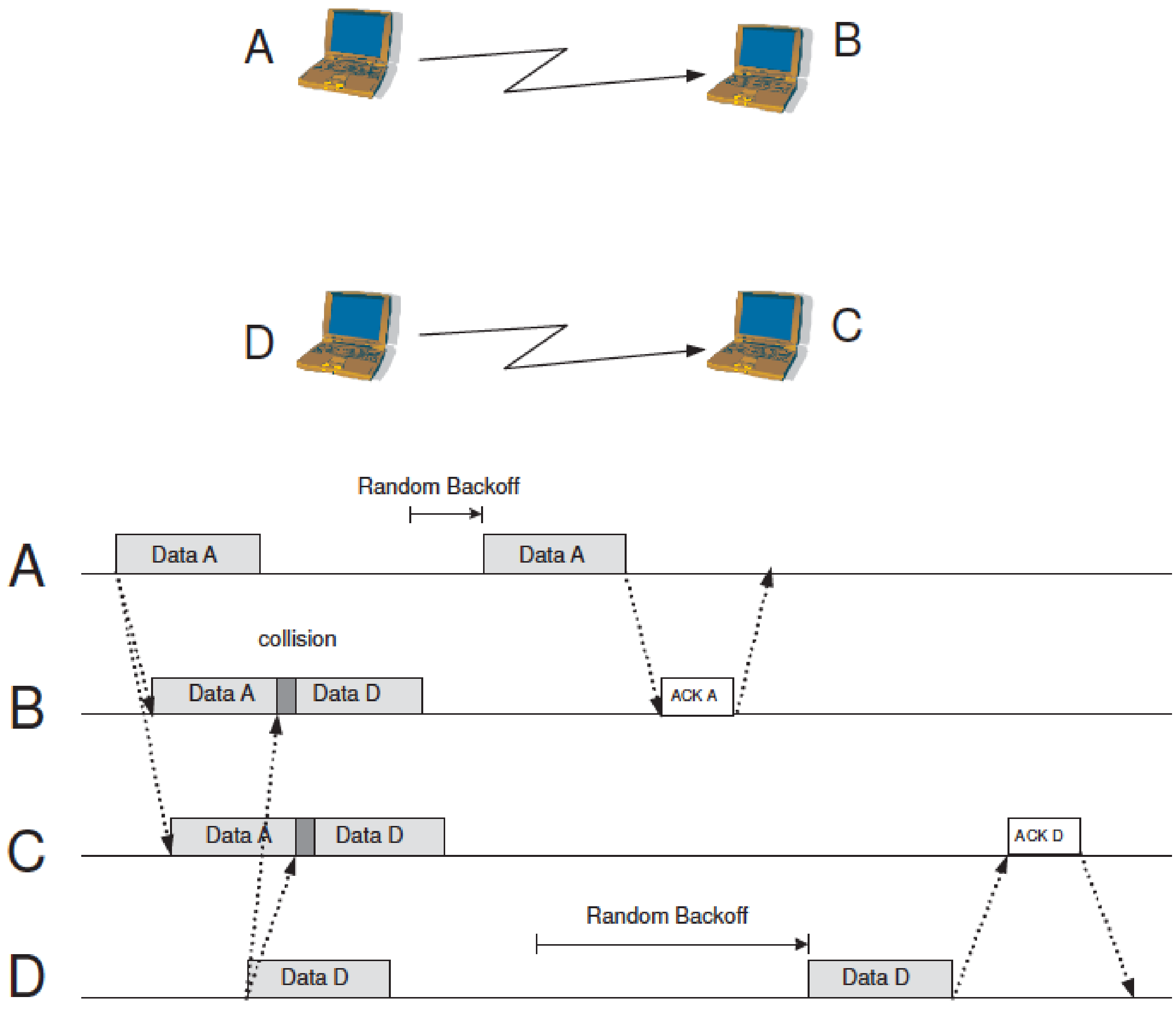
- When node has packet to send
  - transmit at full channel data rate R.
  - no *a priori* coordination among nodes

- two or more transmitting nodes ➜ "collision",



- Collisions are usually resolved by retransmission
- random access MAC protocol specifies:
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed retransmissions)

Collision Resolution

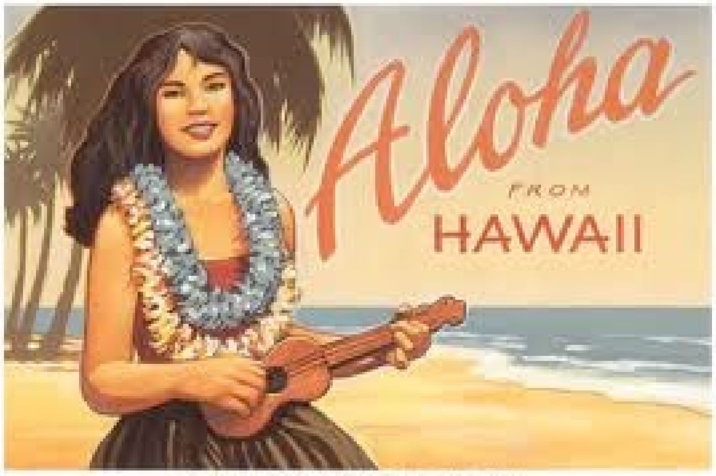


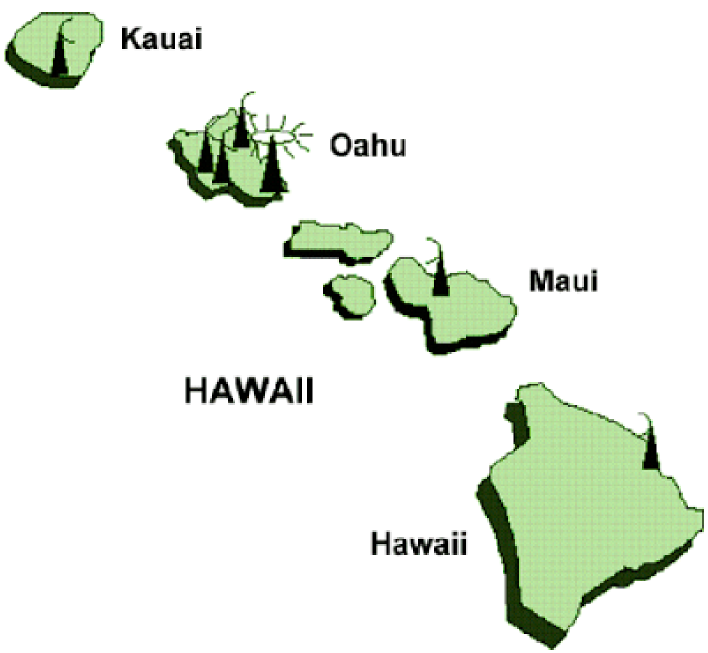
Channel Access

- ALOHA
- Slotted ALOHA
- Carrier Sense Multiple Access (CSMA)
- Carrier Sense Multiple Access with Collision Detection (CSMA-CD) → Ethernet
- Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) → WiFi

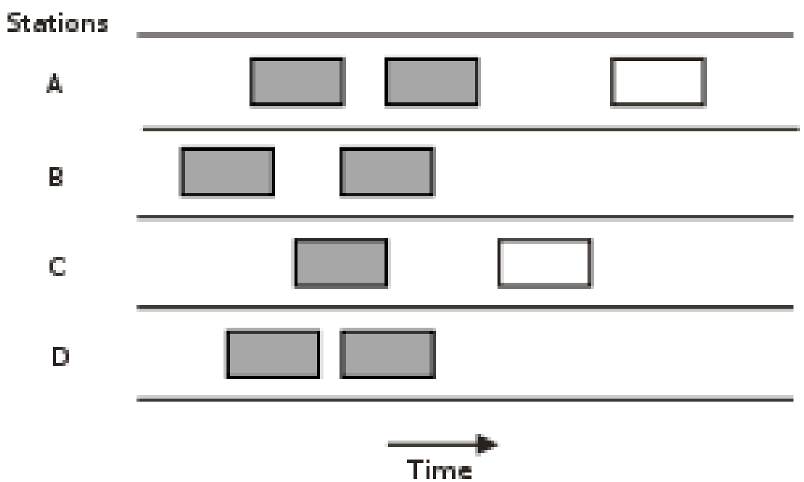
ALOHANet

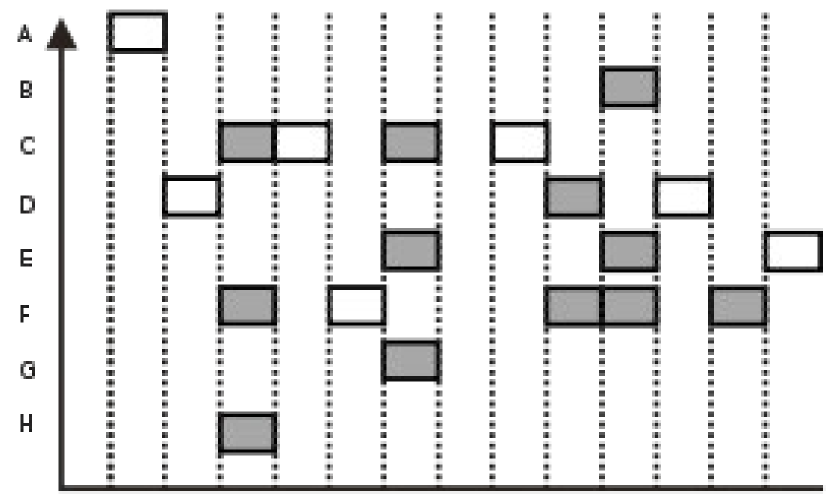
– Developed at the University of Hawaii in 1970
– Goal was to use low-cost commercial radio equipment to connect users on Oahu and the other Hawaiian islands with a central time-sharing computer on the main Oahu campus





Pure and Slotted ALOHA





Slotted ALOHA protocol (shaded slots indicate collision)

Pure (unslotted) ALOHA

Slotted ALOHA

Slotted ALOHA

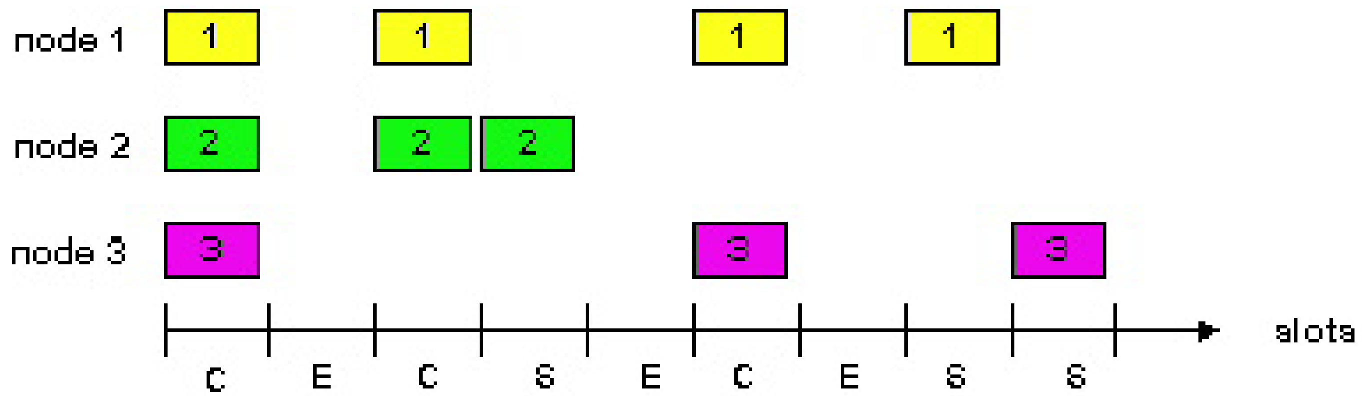
Assumptions:

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

Operation:

- when node obtains new 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

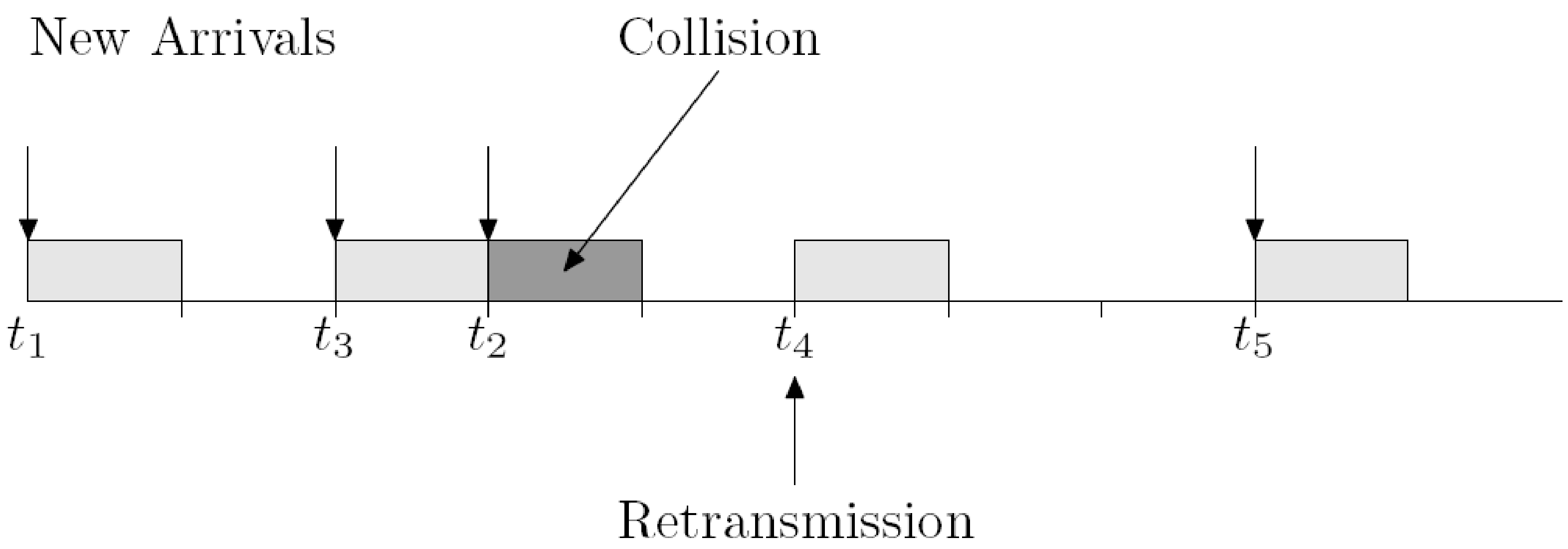
Cons

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

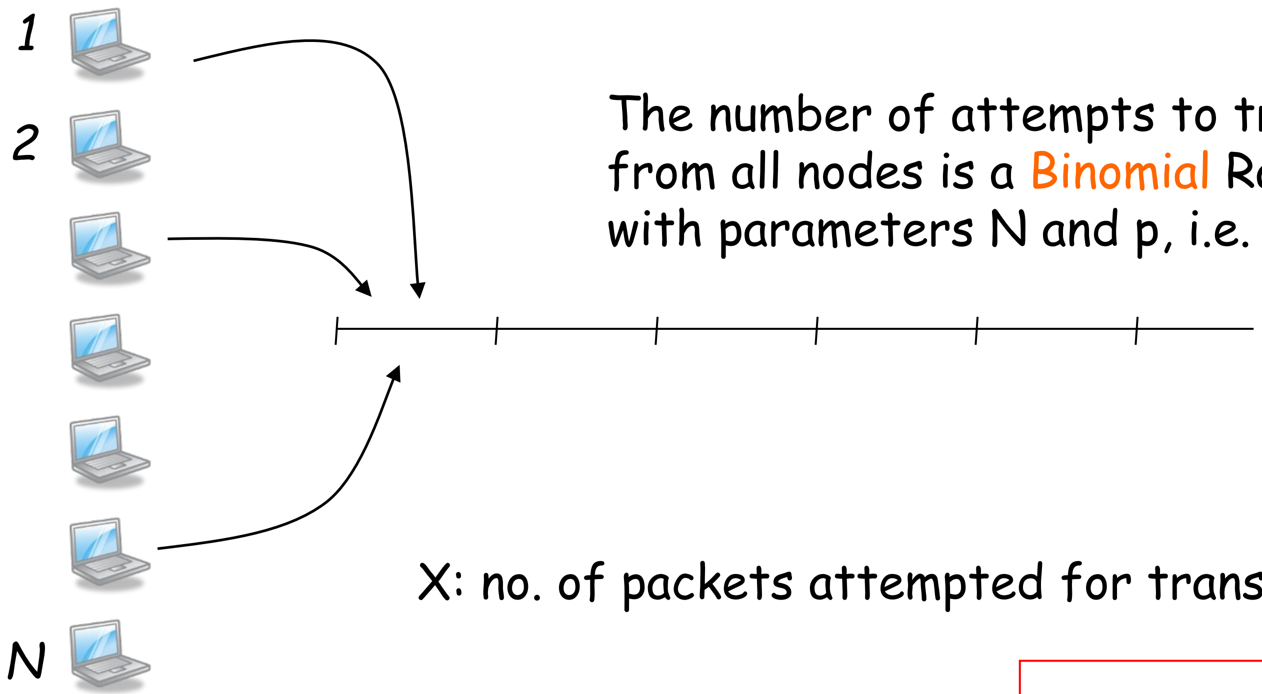
Slotted Aloha efficiency

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*



Binomial Distribution



The number of attempts to transmit a packet from all nodes is a Binomial Random Variable with parameters N and p, i.e. B(N,p)

X: no. of packets attempted for transmission in one timeslot.

$$\boxed{p(X=k)=\binom{N}{k}p^k(1-p)^{N-k}}$$

$$\binom{n}{k}=\frac{n!}{k!(n-k)!}.$$

$$\text{p(idle slot)} = p(X=0)=(1-p)^N$$

$$\text{p(success slot)} = p(X=1)=Np(1-p)^{N-1}$$

$$\text{p(collision slot)} = p(X\geq 2)=1-(1-p)^N-Np(1-p)^{N-1}$$

Time to successful transmission

- prob that there is a successful transmission

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

- prob that a given node (e.g, node 1) has success in a slot

  $Ps = p(1-p)^{N-1}$

- Delay: How long does it take in average for a given node to have its packet successfully transmitted?

- Success in 1 slot Ps,
- Success in 2 slot (1-Ps)Ps,
- Success in k slot $(1-Ps)^{k-1}Ps$,

  - The time (i.e., number of timeslots) to success is Geometrically distributed with parameter Ps. Hence the time to success is 1/Ps.

Example

- A Local Area Network (LAN) uses the slotted ALOHA protocol. There are 20 terminals in the network, and each terminal transmits a packet in a given slot with the probability $p = 0.04$, determine the efficiency of the system.
- Efficiency = Normalized Throughput = Probability that a slot is successful

$$
\begin{aligned}
\text{Normalized Throughput} &= p(X=1) \\
&= Np(1-p)^{N-1} \\
&= 20 \times 0.04 \times (1-0.04)^{19} = 0.37
\end{aligned}
$$

(37% of the total bandwidth)

Example (cont'd)

$N = 20$

– If the packet transmission is not successful due to a collision, the terminal attempts a retransmission in the next slot with probability $p = 0.04$. Determine the expected delay (in time slots) that a packet incurs from the first transmission attempt to the successful reception, assuming that the propagation delay is negligible.

$$P_s = p(1-p)^{N-1} = 0.04 \times 0.96^{19} = 0.0184$$

Geometric Distribution with parameter P_s

$$Delay = \frac{1}{P_s} = \frac{1}{0.0184} = 54.35 \text{ (timeslots)}$$

Example (cont’ d)

– What is the percentage of the slots that are left idle if p = 0.04?

$$\begin{aligned}\text{Idle Slots} &= p(X=0)\\ &= (1-p)^N\\ &= (1-0.04)^{20} = 0.44\end{aligned}$$

Maximum Efficiency

– Find p* that maximizes $Np(1-p)^{N-1}$

$$\frac{d}{dp}\left[Np(1-p)^{N-1}\right] = N(1-p)^{N-1} - N(N-1)p(1-p)^{N-2} = 0 \Rightarrow p^* = \frac{1}{N}$$

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

$$Np^*(1-p^*)^{N-1} = \left(1-\frac{1}{N}\right)^{N-1} \xrightarrow{N\to\infty} e^{-1}$$

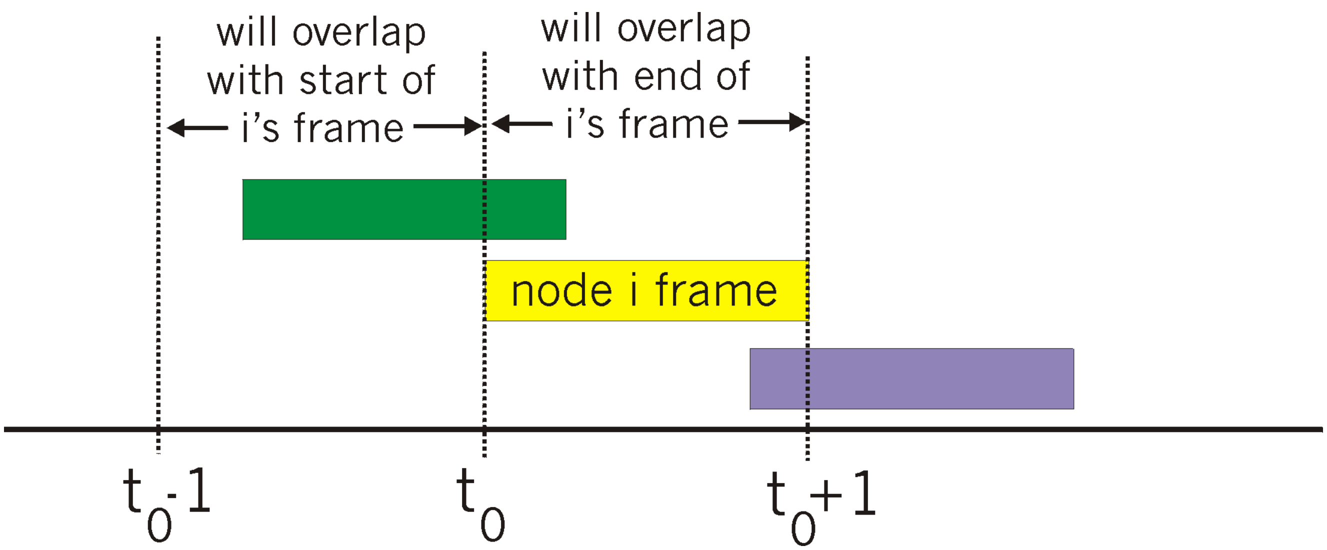
Max efficiency = 1/e = 0.37

> *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 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) $\cdot$

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

P(no other node transmits in $[t_0, t_0+1]$

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

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

– P(success by any node) = $N\,p \cdot (1-p)^{2(N-1)}$

… choosing optimum p and then letting N -> infinity ...

$= 1/(2e) = 0.18$

even worse than slotted Aloha!

Important Concepts

- Multiple access in shared medium
- Possibility for collision of packets
- MAC protocols
  - Channel partitioning
  - Random access
  - Polling (taking turns)
- ALOHA
- Slotted ALOHA
- Efficiency of ALOHA and Slotted ALOHA