BLG 337E- Principles of Computer Communications

Assoc. Prof. Dr. Berk CANBERK

13/11/ 2018

-Medium Access Layer - 3

References:

Data and Computer Communications, William Stallings, Pearson-Prentice Hall, 9th Edition, 2010.

-*Computer Networking, A Top-Down Approach Featuring the Internet*, James F.Kurose, Keith W.Ross, Pearson-Addison Wesley, 6th Edition, 2012.

-Google!

CSMA/CD (collision detection)

CSMA/CD: carrier sensing, deferral as in CSMA

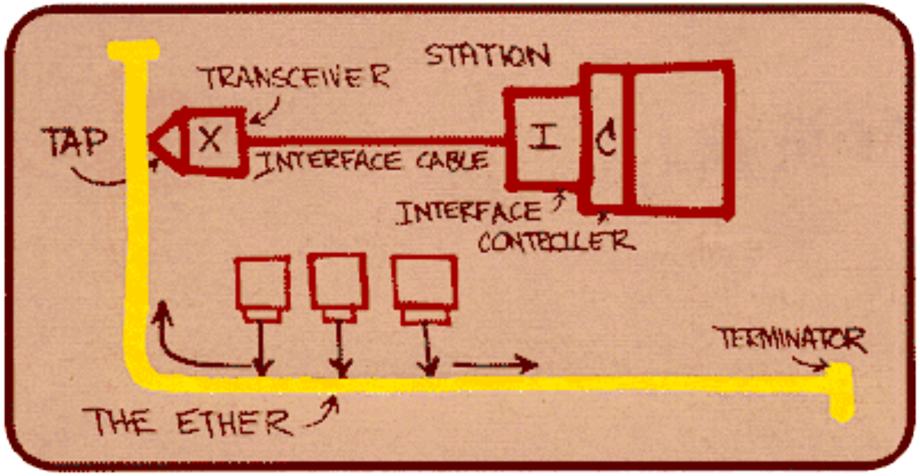
- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage

- 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

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 *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

Ethernet



Original picture drawn by Bob Metcalfe, inventor of Ethernet (1972 – Xerox PARC)

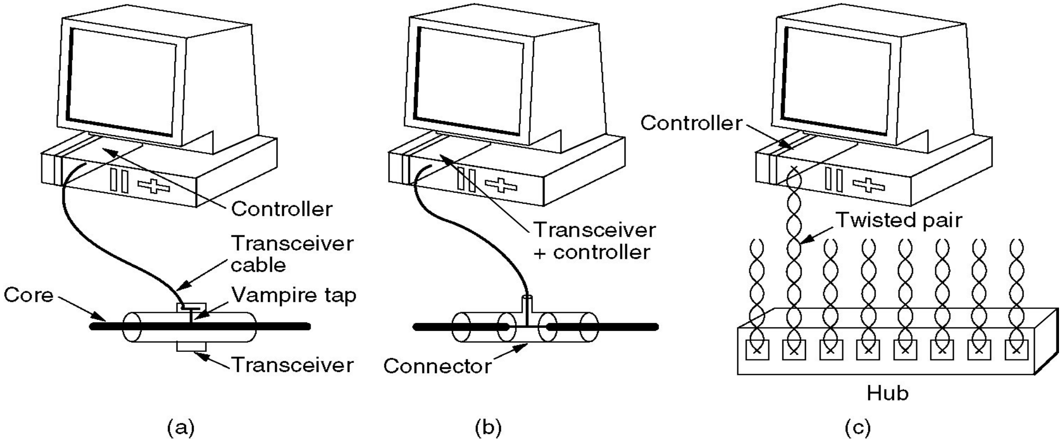
- Developed by Bob Metcalfe and others at Xerox PARC in mid-1970s
  - Roots in Aloha packet-radio network
  - Standardized by Xerox, DEC, and Intel in 1978
  - LAN standards define MAC and physical layer connectivity
    - IEEE 802.3 (CSMA/CD - Ethernet) standard – originally 2Mbps
    - IEEE 802.3u standard for 100Mbps Ethernet
    - IEEE 802.3z standard for 1,000Mbps Ethernet
- CSMA/CD: Ethernet’s Media Access Control (MAC) policy (1-persistent CSMA/CD with binary exponential backoff)
- Bandwidths: 10Mbps, 100Mbps, 1Gbps
- Max bus length: 2500m: 500m segments with 4 repeaters
- Bus and Star topologies are used to connect hosts
- Manchester Encoding

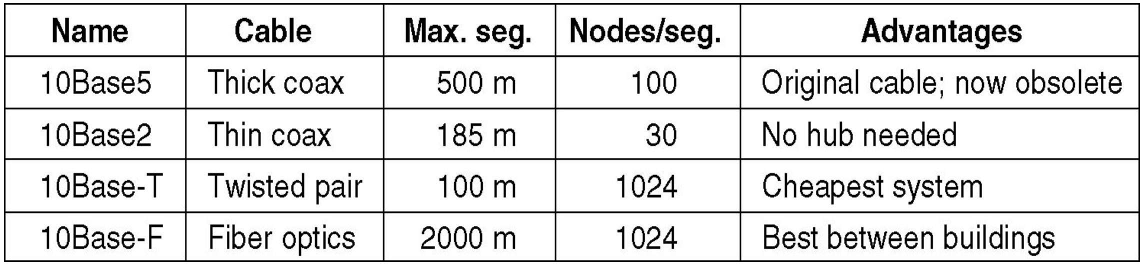
Ethernet Cabling

10Base-T:

- Unshielded twisted pair (UTP) medium
- Star-shaped topology
  - Stations connected to central point (hub), (multiport repeater)
  - Two twisted pairs (transmit and receive)
  - Repeater accepts input on any one line and repeats it on all other lines
- Link limited to 100 m on UTP
- Multiple levels of hubs can be cascaded

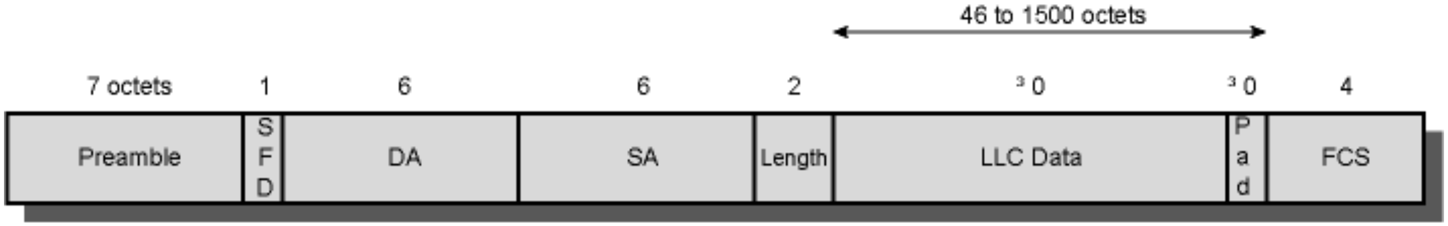
(a) 10Base5, (b) 10Base2, (c) 10Base-T.





Name	Cable	Max. seg.	Nodes/seg.	Advantages
10Base5	Thick coax	500 m	100	Original cable; now obsolete
10Base2	Thin coax	185 m	30	No hub needed
10Base-T	Twisted pair	100 m	1024	Cheapest system
10Base-F	Fiber optics	2000 m	1024	Best between buildings

802.3 Ethernet Frame Format



SFD = Start of frame delimiter
DA = Destination address
SA = Source address
FCS = Frame check sequence

- Sending adapter encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame
  - Preamble (+SFD): 7 bytes with pattern 10101010 followed by 1 byte with pattern 10101011 used to synchronize receiver
  - Start of Frame Delimiter (SFD): indicates start of frame (1 byte with pattern 10101011)
  - Addresses: 6 bytes, frame is received by all adapters on a LAN and dropped if address does not match, globally unique address assigned by manufacturer, e.g. 8:0:e4:b1:2
  - Length: frame size
  - Pad: Zeroes used to ensure ***minimum frame length of 64 Bytes (WHY??)***
  - FCS (CRC) Cyclic Redundancy Check: check sequence to detect bit errors, if error is detected, the frame is simply dropped
  - Body can contain up to 1500 bytes of data

CSMA/CD (Collision Detection)

- CSMA an improvement over ALOHA because no station transmits when it senses the channel busy

- Another improvement: *stations abort their transmissions as soon as they detect a collision*
  - if two stations sense the channel idle and begin transmission simultaneously they will both detect the collision immediately
  - Rather than finishing transmitting their frames, which will be corrupted, stop transmitting frames as soon as collision detected

    Quickly terminating damaged frames saves time and bandwidth!

- This protocol is called CSMA/CD (Carrier Sense Multiple Access with Collision Detection).

CSMA/CD (Collision Detection)

CSMA/CD: carrier sensing, deferral as in CSMA

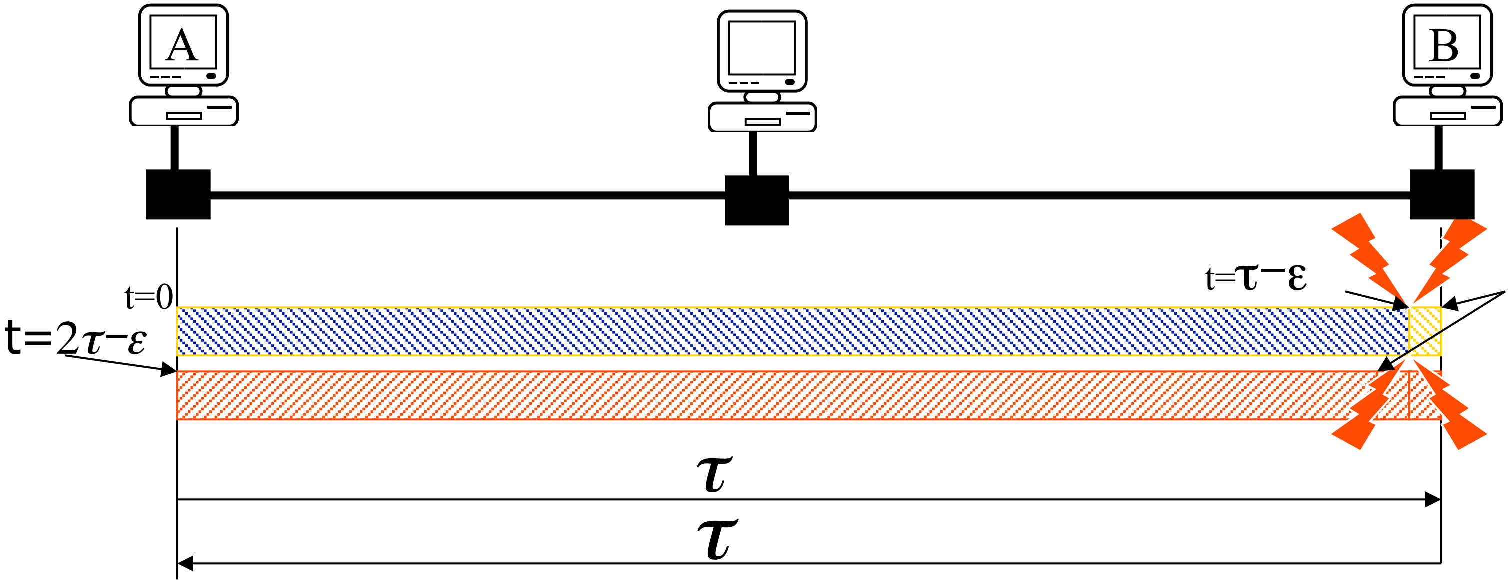
- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage
- persistent or non-persistent retransmission

- Collision detection:
  - easy in wired LANs: measure signal strengths (power or pulse width), compare transmitted, received signals
  - difficult in wireless LANs: receiver shut off while transmitting

CSMA/CD

Time to Detect Collision

- Let the time for a signal to propagate between two farthest stations be τ
- It takes 2τ seconds for two stations to realize that there has been a collision after starting the transmission



Events:

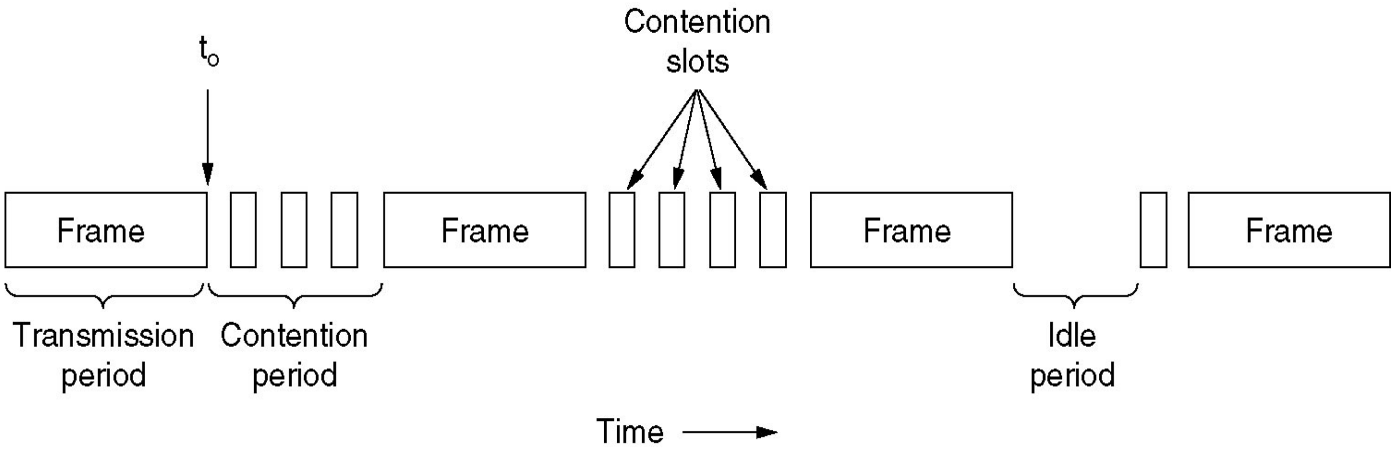
t=0: Host A starts transmitting a packet.

t=τ−ε Just before the first bit reaches Host B, Host B senses the line to be idle and starts to transmit a packet.
A collision takes place near Host B.

t=2τ−ε: Host A receives the noise burst caused by the collision

CSMA/CD

CSMA/CD can be in one of three states: contention, transmission, or idle.



- The minimum time it takes to detect a collision is just the time it takes for the signal to propagate from any computer to any other computer and back again → 2τ Slot Time
  - E.g., for a 1km long cable: τ=(1000 m)/ (2x10^8 m/sec)=5 μsec

Contention Period: modeled as a Slotted ALOHA System with Slot Time of 2τ

"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

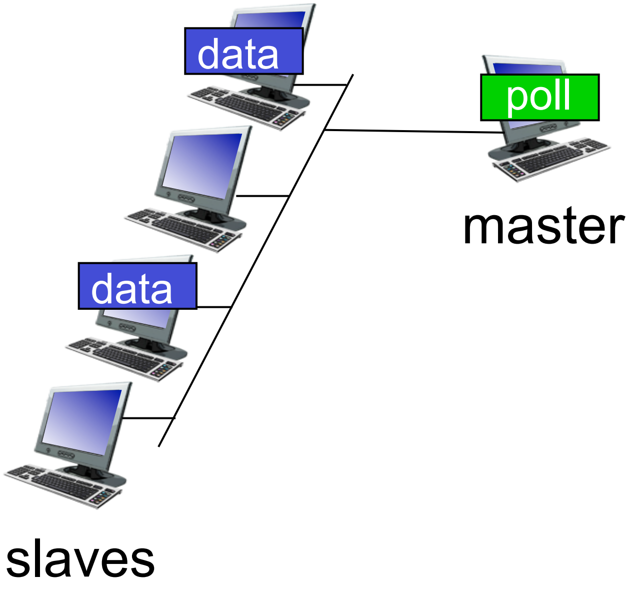
"taking turns" protocols

look for best of both worlds!

"Taking turns" MAC protocols

polling:

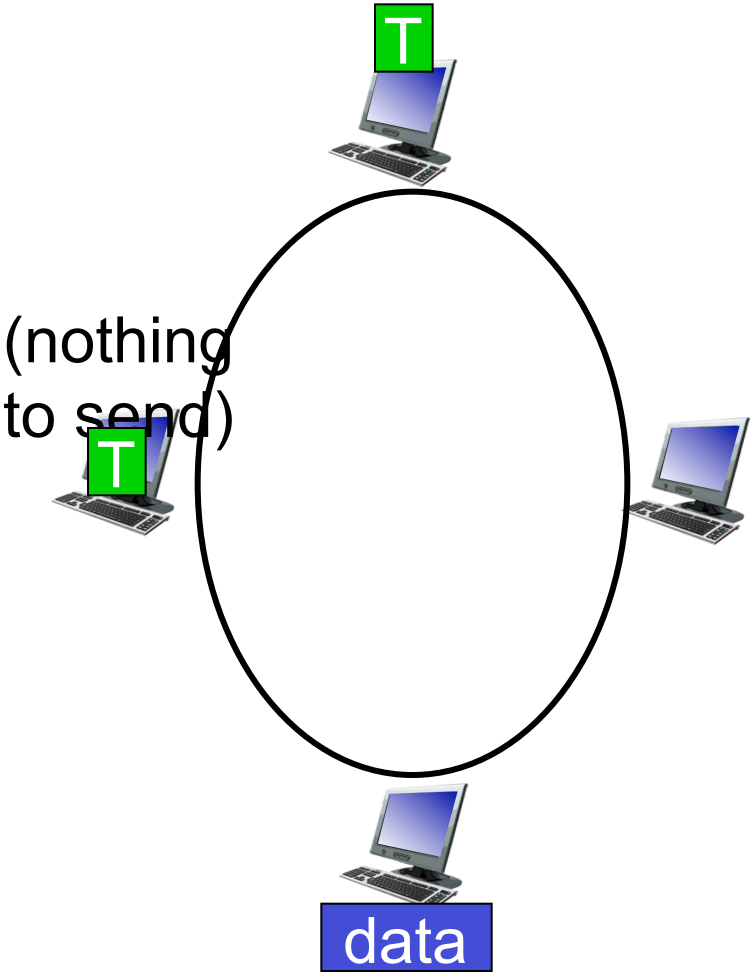
- master node "invites" slave nodes to transmit in turn
- typically used with "dumb" slave devices
- concerns:
  - polling overhead
  - latency
  - single point of failure (master)



"Taking turns" MAC protocols

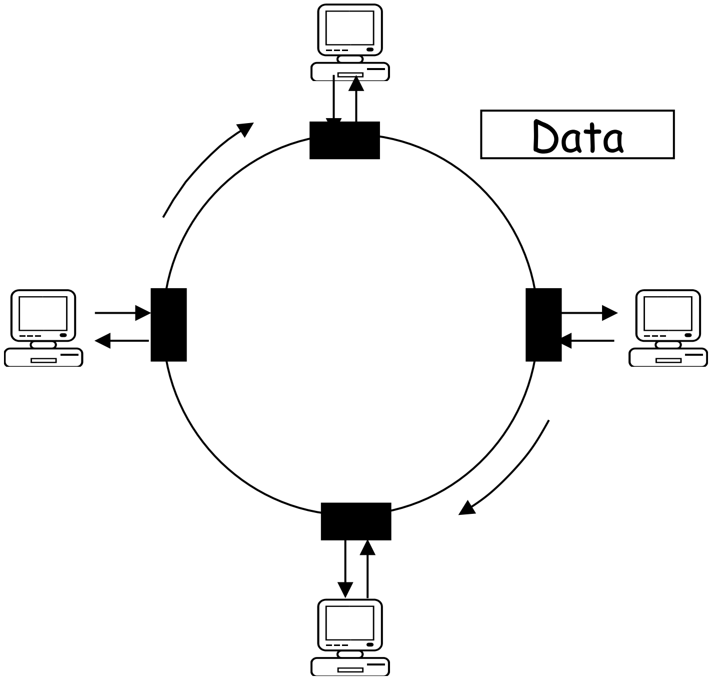
token passing:

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

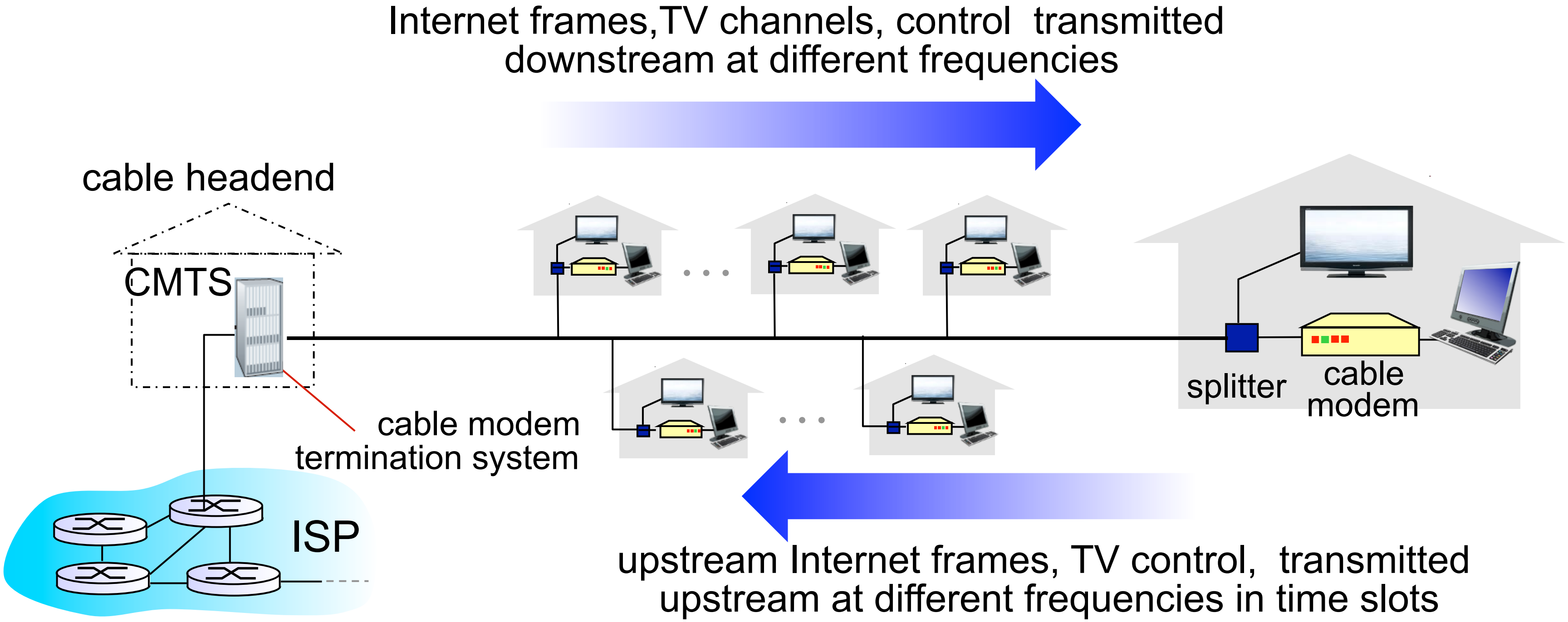


A Quick Word about Token Ring

- Developed by IBM in early 80’s as a new LAN architecture
  - Consists of nodes connected into a ring (typically via concentrators)
  - Special message called a token is passed around the ring
    - When nodes gets the token it can transmit for a limited time
    - Every node gets an equal opportunity to send
  - IEEE 802.5 standard for Token Ring
- Designed for predictability, fairness and reliability
  - Originally designed to run at either 4Mbps and 16Mbps
- Still used and sold but beaten out by Ethernet

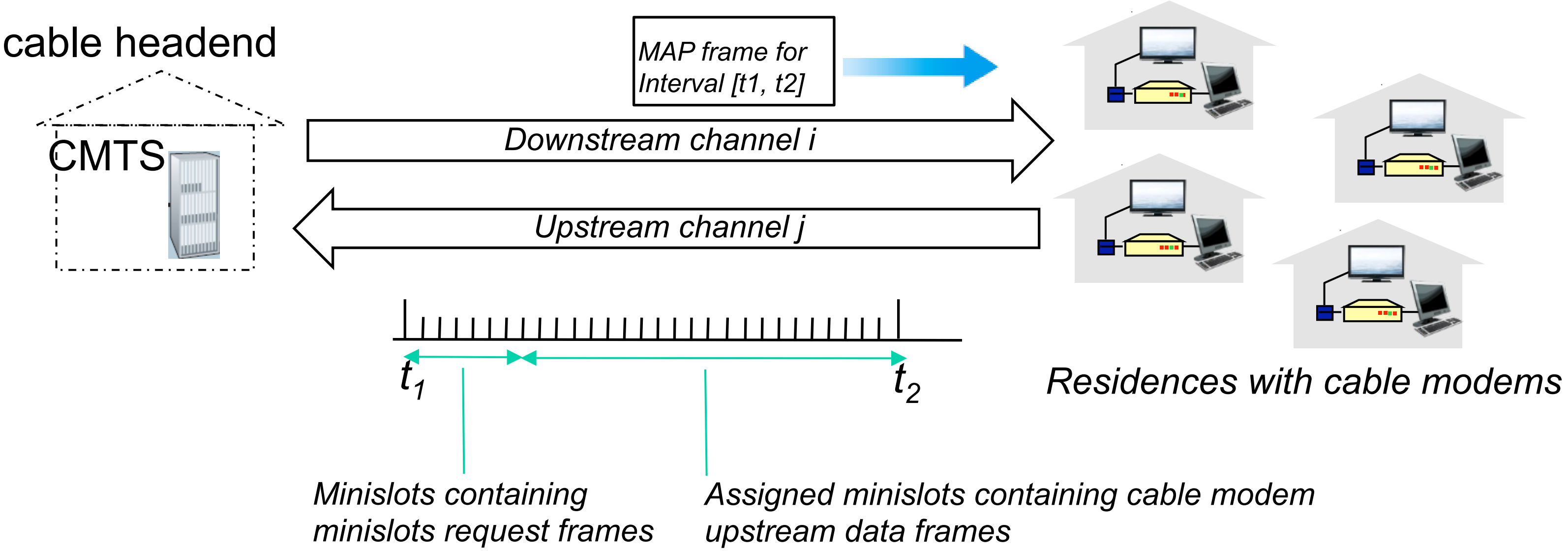


Cable access network


Internet frames,TV channels, control transmitted downstream at different frequencies
cable headend
CMTS
cable modem termination system
ISP
splitter
cable modem
upstream Internet frames, TV control, transmitted upstream at different frequencies in time slots

- multiple 40Mbps downstream (broadcast) channels
  - single CMTS transmits into channels
- multiple 30 Mbps upstream channels
  - multiple access: all users contend for certain upstream channel time slots (others assigned)

Cable access network


cable headend
CMTS
MAP frame for Interval [t1, t2]
Downstream channel i
Upstream channel j
t_1
t_2
Residences with cable modems
Minislots containing minislots request frames
Assigned minislots containing cable modem upstream data frames

DOCSIS: data over cable service interface spec

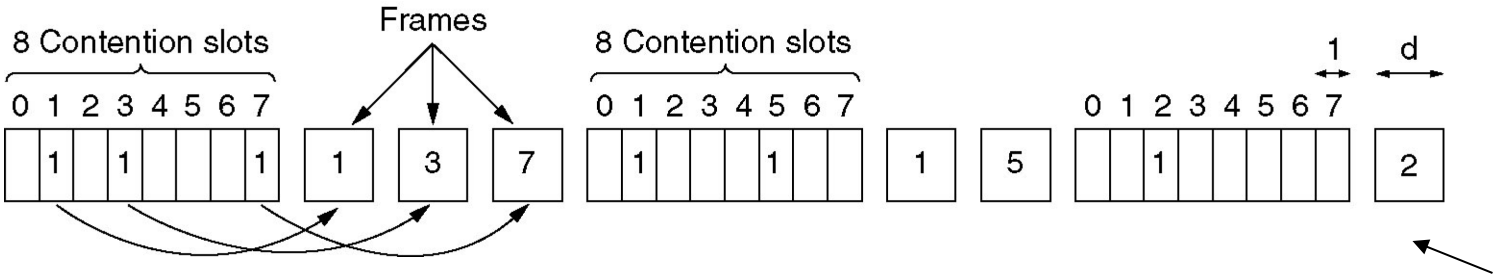
- FDM over upstream, downstream frequency channels
- TDM upstream: some slots assigned, some have contention
  - downstream MAP frame: assigns upstream slots
  - request for upstream slots (and data) transmitted random access (binary backoff) in selected slots

Collision-Free Protocols

- Collisions can still occur with CSMA/CD
- Collisions adversely affect the system performance
- Effect is greater when the cable is long (large τ)
- Resolve contention without any collision!
    - <u>Collision-free protocols:</u> Bit-Map Protocol, Binary Countdown Protocol etc.

Bit-Map Protocol

- Assume that there are N stations each with unique address from 0 to N-1
- Each contention period consists of exactly N slots
  - If station 0 has a frame to send, it transmits a 1 bit during 0th slot
  - No other station is allowed to transmit during this slot
  - Station j may announce that it has a frame to send (only if so) by sending 1 bit in jth slot
  - After all N slots have passed by, each station knows which station will transmit
  - They begin transmitting in the numerical order as agreed before→ NO COLLISION!



If a station is ready just after its bit slot has passed by, it must wait until the bitmap has come around again!!

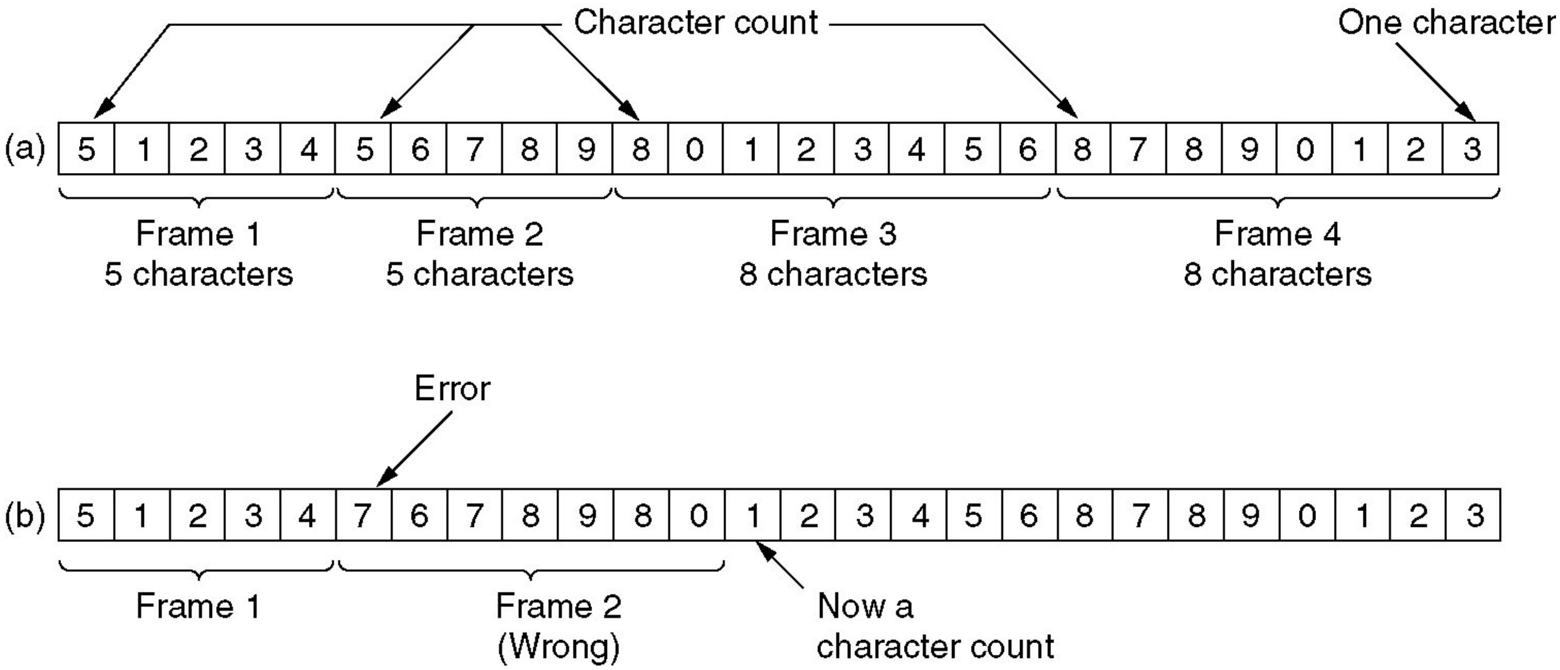
Reservation protocol !

Framing

- DLL breaks bit stream into discrete frames
- Computes checksum of each frame
- Start + end of frame determination:
  - Character count
  - Start/end characters with character stuffing
  - Start/end flags with bit stuffing
  - Physical layer coding violations

Framing (Character Count)

A character stream. (a) Without errors. (b) With one error.



Character count

One character

(a) 5 1 2 3 4 5 6 7 8 9 8 0 1 2 3 4 5 6 8 7 8 9 0 1 2 3

Frame 1
5 characters

Frame 2
5 characters

Frame 3
8 characters

Frame 4
8 characters

Error

(b) 5 1 2 3 4 7 6 7 8 9 8 0 1 2 3 4 5 6 8 7 8 9 0 1 2 3

Frame 1

Frame 2
(Wrong)

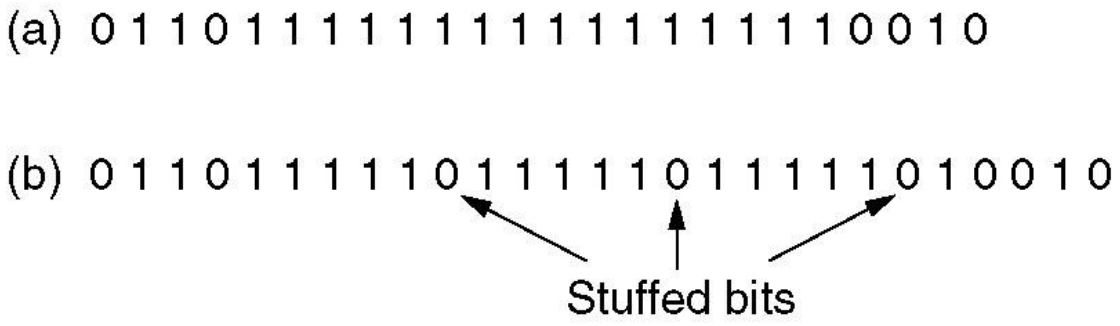
Now a
character count

Problem occurs when control field is corrupted!

Framing (Bit Stuffing)

- Each frame begins and ends with a special bit pattern called a flag byte [01111110].
- Whenever sender data link layer encounters *five consecutive ones* in the data stream, it automatically stuffs a 0 bit into the outgoing stream.
- When the receiver sees *five consecutive incoming ones followed by a 0 bit*, it automatically destuffs the 0 bit before sending the data to the network layer.

Framing (Bit Stuffing)



(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

Stuffed bits

(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Bit stuffing

(a) The original data.

(b) The data as they appear on the line.

(c) The data as they are stored in receiver’s memory after destuffing.

Error Control

- Applications require certain reliability level
  - Data applications require error-free transfer
  - Voice & video applications tolerate some errors
- Transmission errors exist
  - Single bit errors
  - Burst errors (which one is better???)
  - Lost frames vs. Damaged frames (when??)
- ***Error detection***
  - Error-detecting codes: CRC, checksum etc.
  - Would suffice (along with a retransmission-based strategy) in relatively reliable channels
- ***Error correction***
  - Error-correcting codes: Hamming codes…
  - Required for error-prone channels
- Two basic approaches:
  - Error ***detection*** & retransmission (ARQ: Automatic Repeat reQuest)
  - Forward error ***correction*** (FEC)

Error Control

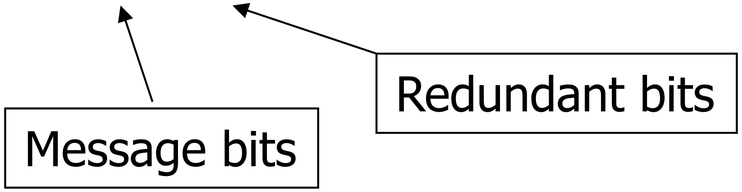
- Error rate
  - Bit error rate (BER) : probability of a transmitted bit being received wrong
    - e.g., 10^{-7} for satellite, 10^{-9} for MW, 10^{-11} for fiber
  - Packet/frame error rate
- For a given BER and frame length n

P[frame correct]= $(1\text{-BER})^n$

P[frame has error]=1- $(1\text{-BER})^n \cong$ n x BER

Error Detection

- To detect or correct errors
  - Additional (redundant) bits added by transmitter to the original data to form codewords
  - Codeword length $n=m+r$



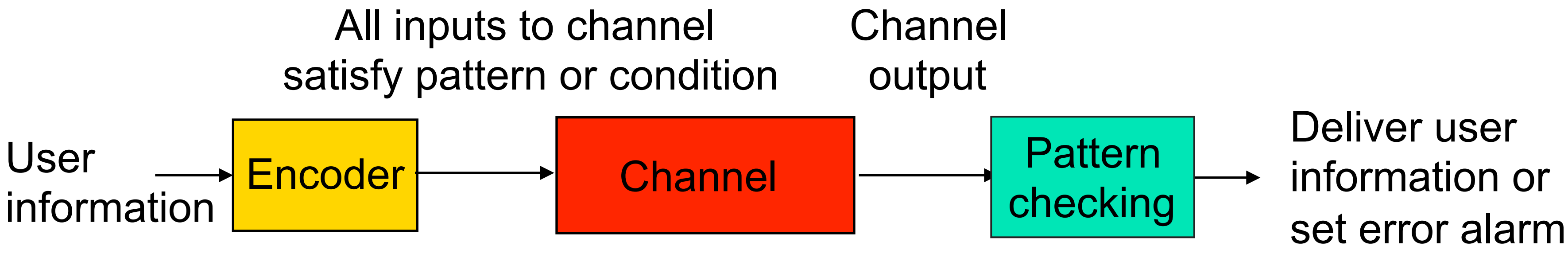
Message bits

Redundant bits

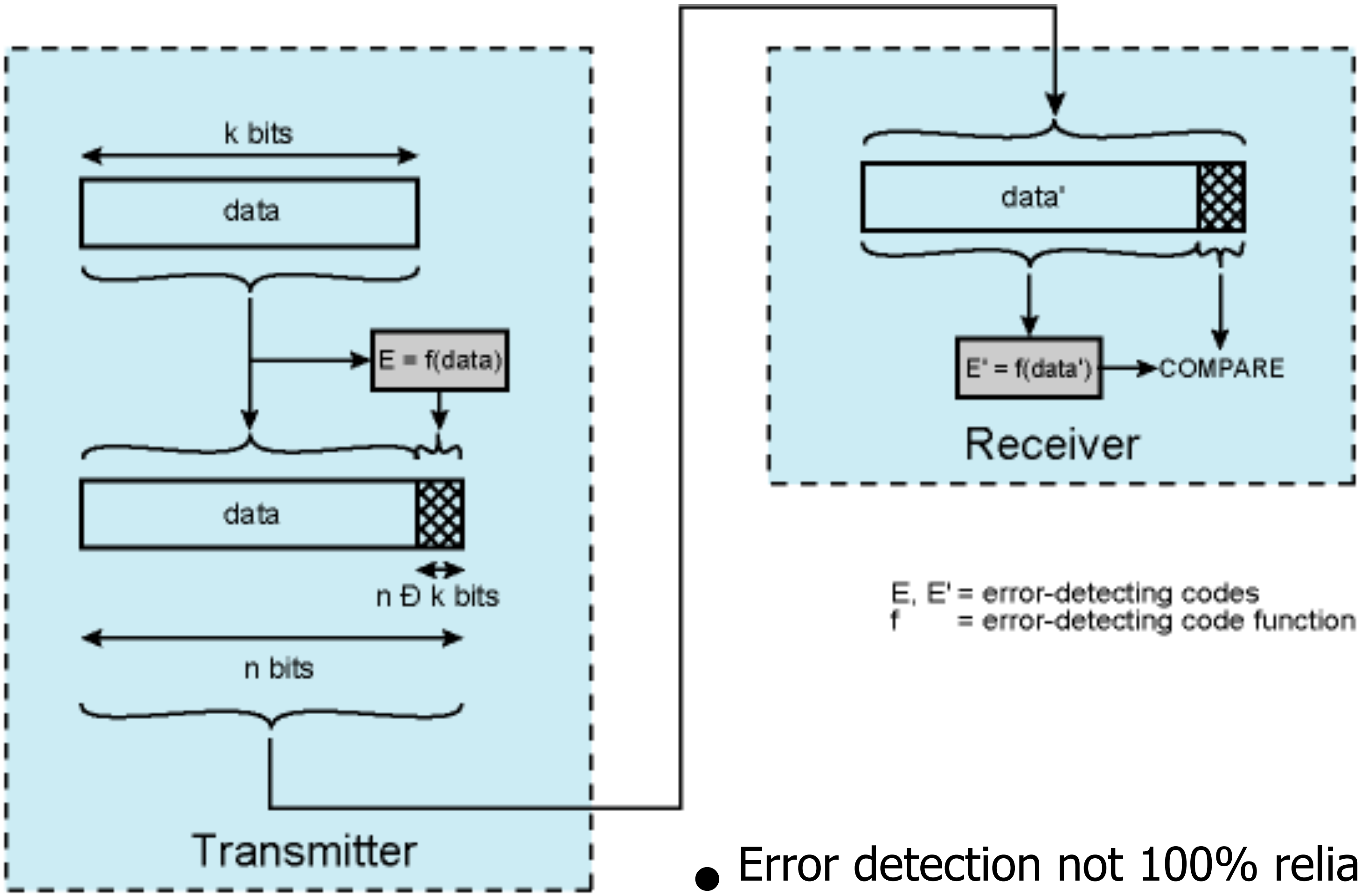
  - e.g. Parity
    - Value of parity bit is such that character has even (even parity) or odd (odd parity) number of ones

General Idea

- All transmitted data blocks ("codewords") satisfy a pattern
- If received block does not satisfy pattern, it is in error
- Redundancy: Only a subset of all possible blocks can be codewords
- Blindspot: when channel transforms a codeword into another codeword


All inputs to channel satisfy pattern or condition
Channel output
User information
Encoder
Channel
Pattern checking
Deliver user information or set error alarm

Error Detection Process



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

Single Parity Check

- Append an overall parity check to *k* information bits

Info Bits: $b_1, b_2, b_3, \ldots, b_k$

Check Bit: $b_{k+1} = b_1 + b_2 + b_3 + \ldots + b_k$ modulo 2

Codeword: $(b_1, b_2, b_3, \ldots, b_{k,}, b_{k+1})$

- All codewords have even # of 1s
- Receiver checks to see if # of 1s is even
  - All error patterns that change an odd # of bits are detectable
- Parity bit used in ASCII code

All even-numbered error patterns are undetectable!!!

Example of Single Parity Code

- Information (7 bits): (0, 1, 0, 1, 1, 0, 0)
- Parity Bit: b_8 = 0 + 1 +0 + 1 +1 + 0 = 1
- Codeword (8 bits): (0, 1, 0, 1, 1, 0, 0, 1)

- If single error in bit 3 : (0, 1, 1, 1, 1, 0, 0, 1)
  - # of 1's =5, odd
  - Error detected

- If errors in bits 3 and 5: (0, 1, 1, 1, 0, 0, 0, 1)
  - # of 1's =4, even
  - Error not detected

Other Error Detection Codes

- Many applications require very low error rate
- Need codes that detect the vast majority of errors
- Single parity check codes do not detect enough errors
- The following error detecting codes used in practice:
  - CRC Polynomial Codes
  - Internet Check Sums (at Transport Layer)

Polynomial Codes

- Polynomials for codewords and polynomial arithmetic
- Implemented using shift-register circuits
- Also called *cyclic redundancy check (CRC)* codes
- Most data communications standards use polynomial codes for error detection
    - For a block of *k* bits transmitter generates *n* bit sequence
    - Transmit *k+n* bits which is exactly divisible by some number
    - Receiver divides frame by that number
        - If no remainder, assume no error

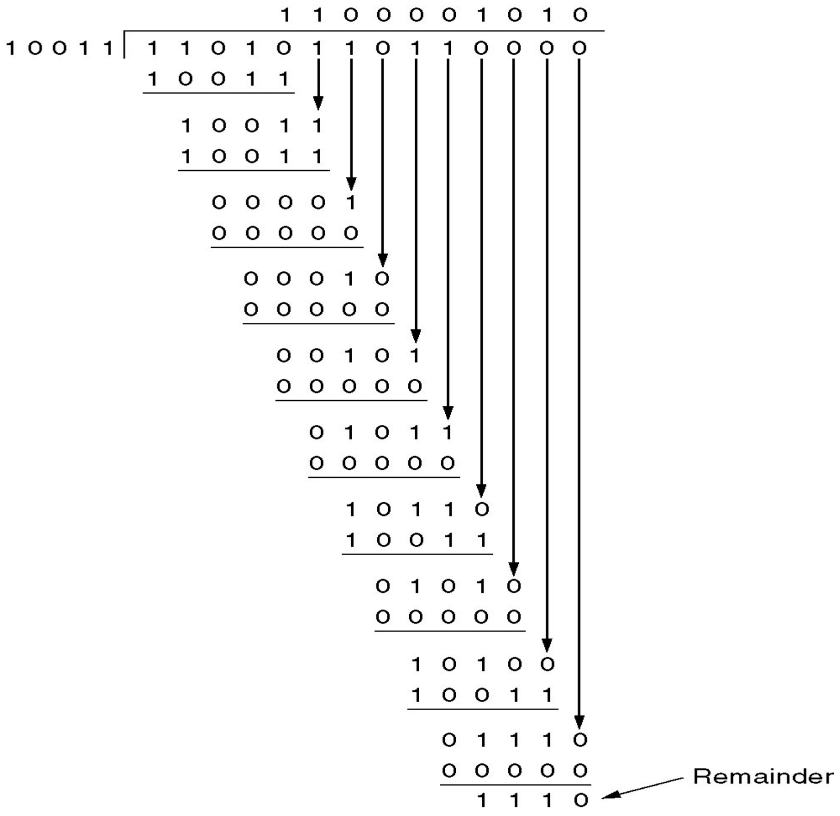
Error-Detecting Codes (CRC)

- 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 (ATM, HDLC)

Divide D·2^r by G, get remainder R,
and transmit <D,R>

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

Frame : 1 1 0 1 0 1 1 0 1 1
Generator: 1 0 0 1 1
Message after 4 zero bits are appended: 1 1 0 1 0 1 1 0 1 1 0 0 0 0



```
                    1 1 0 0 0 0 1 0 1 0
1 0 0 1 1 | 1 1 0 1 0 1 1 0 1 1 0 0 0 0
            1 0 0 1 1
              1 0 0 1 1
              1 0 0 1 1
                0 0 0 0 1
                0 0 0 0 0
                  0 0 0 1 0
                  0 0 0 0 0
                    0 0 1 0 1
                    0 0 0 0 0
                      0 1 0 1 1
                      0 0 0 0 0
                        1 0 1 1 0
                        1 0 0 1 1
                          0 1 0 1 0
                          0 0 0 0 0
                            1 0 1 0 0
                            1 0 0 1 1
                              0 1 1 1 0
                              0 0 0 0 0
                                1 1 1 0   Remainder
```

Transmitted frame: 1 1 0 1 0 1 1 0 1 1 1 1 1 0

Standard Generator Polynomials

CRC = cyclic redundancy check

- CRC-8:

  $= x^8 + x^2 + x + 1$ ATM

- CRC-16:

  $= x^{16} + x^{15} + x^2 + 1$ Bisync

  $= (x + 1)(x^{15} + x + 1)$

- CCITT-16:

  $= x^{16} + x^{12} + x^5 + 1$ HDLC, XMODEM, V.41

- CCITT-32: IEEE 802, DoD, V.42

  $= x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$