Link layer
Error detection and correction

CE 352, Computer Networks

Salem Al-Agtash

Lecture 19

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

Recall (Layers)

- Protocols, reference model, layers
- Internet protocol stack

Today

- Data link layer
  - Error detection and correction
  - Multiple access protocol – broadcast channels: Aloha, CSMA/CD
  - Link layer addressing: MAC and ARP
  - Local area networks: Ethernet, VLANS

		Bits
Application	Web Browser (HTTP) Email (SMTP, IMAP, POP3) Remote Login (Telnet) Name Resolution (DNS) File Transfer (FTP, TFTP)	
Transport	Transmission Control Protocol (TCP) User Datagram Protocol (UDP)	Port: 16
Network	Internet Protocol IP OSPF, BGP, SNMP, ICMP	IPv4: 32 OR IPv6: 128
Data Link	Error detection/correction Ethernet and VLAN MAC Protocols: Aloha, CSMA	MAC : 48
Physical	Copper Twisted Pair Fiber Optic Radio	

Terminology

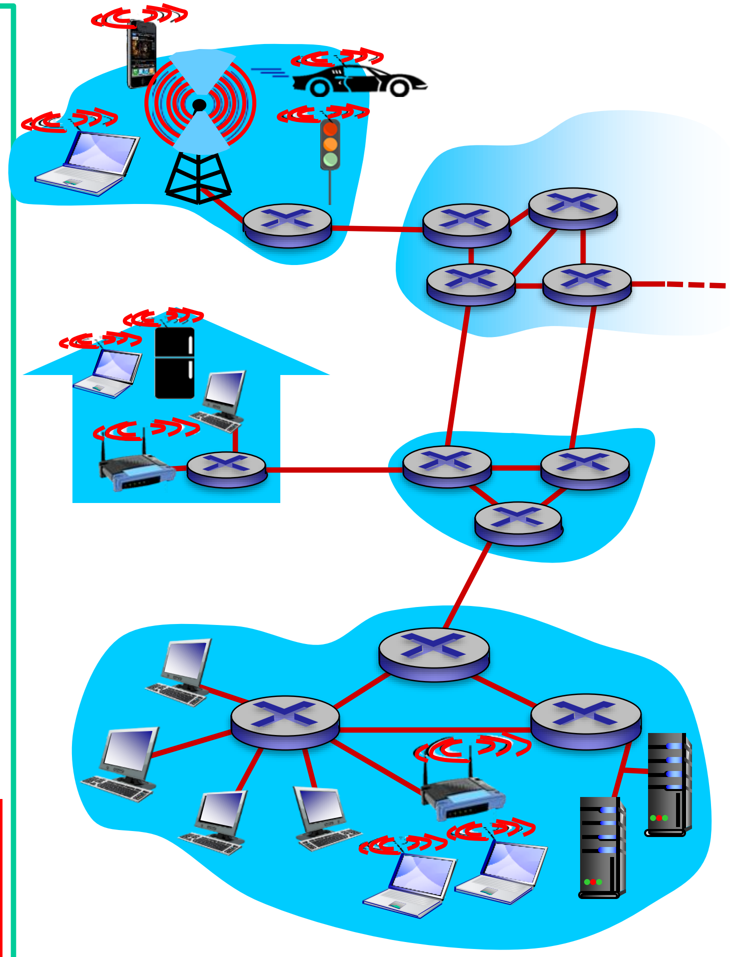
nodes: Hosts and routers

links: communication channels that connect adjacent nodes along communication path

- wired links
- wireless links
- WANs, 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



Context

Datagram transferred by different link protocols over different links:

- e.g., Ethernet on first link (LAN), fiber (SONET, SDH) intermediate links (WAN), 802.11 on last link (LAN)

Each link protocol provides different services

- e.g., may or may not provide rdt over link

transportation analogy:

trip from Princeton to Lausanne

- limo: New York to JFK
- plane: JFK to Geneva
- train: Geneva to Lausanne

tourist = datagram

transport segments = communication links

transportation mode = link layer protocol

travel agent = routing algorithm

Link layer services

framing, link access:

- encapsulate datagram into frame, adding header, trailer
- channel access at shared medium – Multiple Access Protocols
- "MAC" addresses used in frame headers to identify source, destination
  - different from IP address!

reliable delivery between adjacent nodes

- Stop-and-wait, Sliding Window
- Seldom used on low bit-error link (fiber, some twisted pair)
- Wireless links: high error rates

Link layer services (more)

flow control:

- pacing between adjacent sending and receiving nodes

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

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?

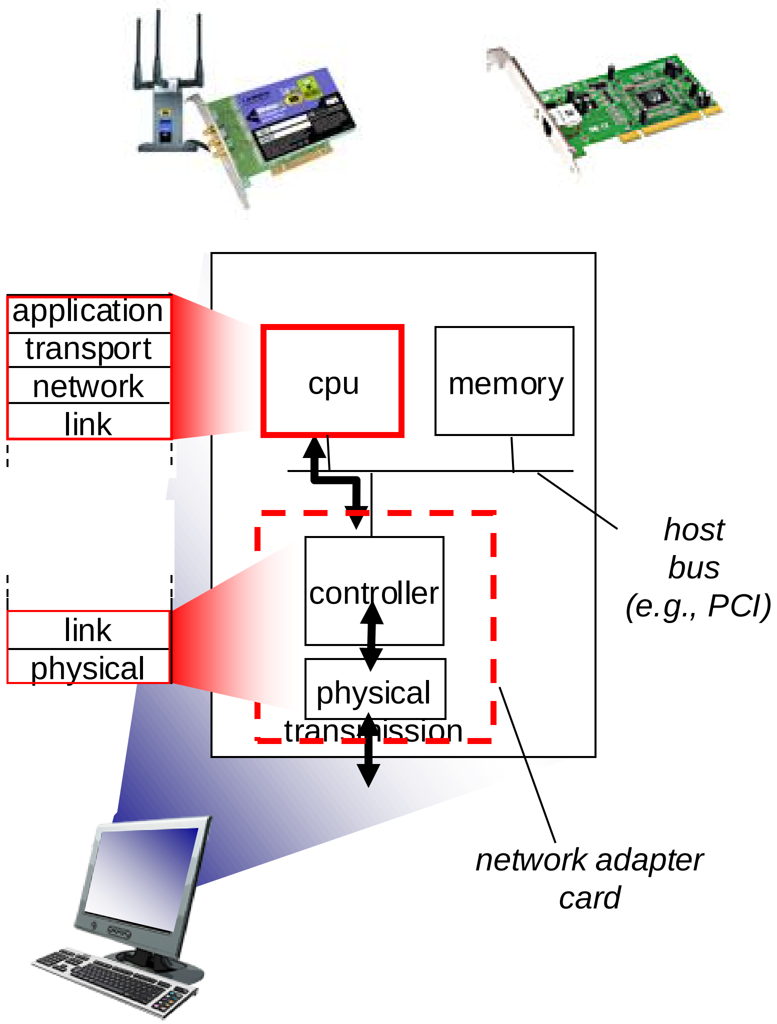
in each and every host

link layer implemented in "adaptor" (aka *network interface card* NIC) or on a chip

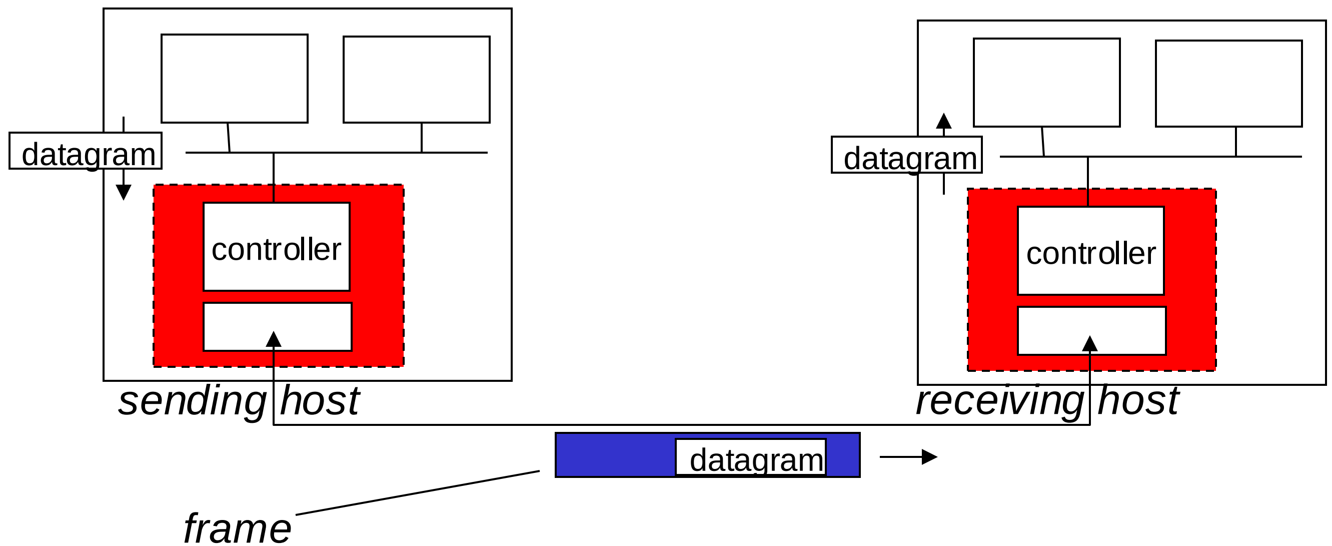
- Ethernet card, 802.11 card; Ethernet chipset
- implements link, physical layer

attaches into host's system buses

combination of hardware, software, firmware



Adaptors communicating



sending side:

- encapsulates datagram in frame
- adds error checking bits, flow control, etc.

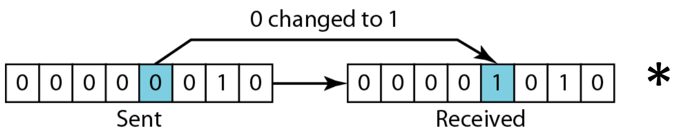
receiving side

- looks for errors, flow control, etc.
- extracts datagram, passes to upper layer at receiving side

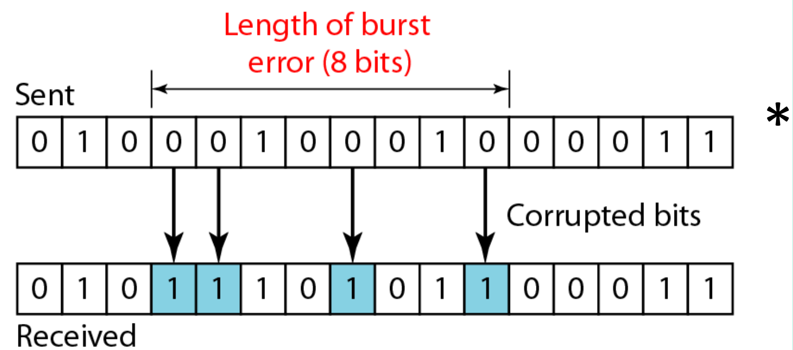
Transmission errors

Data transmission can contain errors

- Single-bit



- Burst errors of length n
  (n: distance between the first and
  last errors in data block)



How to detect/ correct errors

- If only data is transmitted, errors cannot be detected/ corrected
  - Send more information with data that satisfies a special relationship
  - Add redundancy

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

Error detecting/ correcting codes

Error-Detecting Codes

- Parity.
- Checksums.
- Cyclic Redundancy Checks (CRCs).

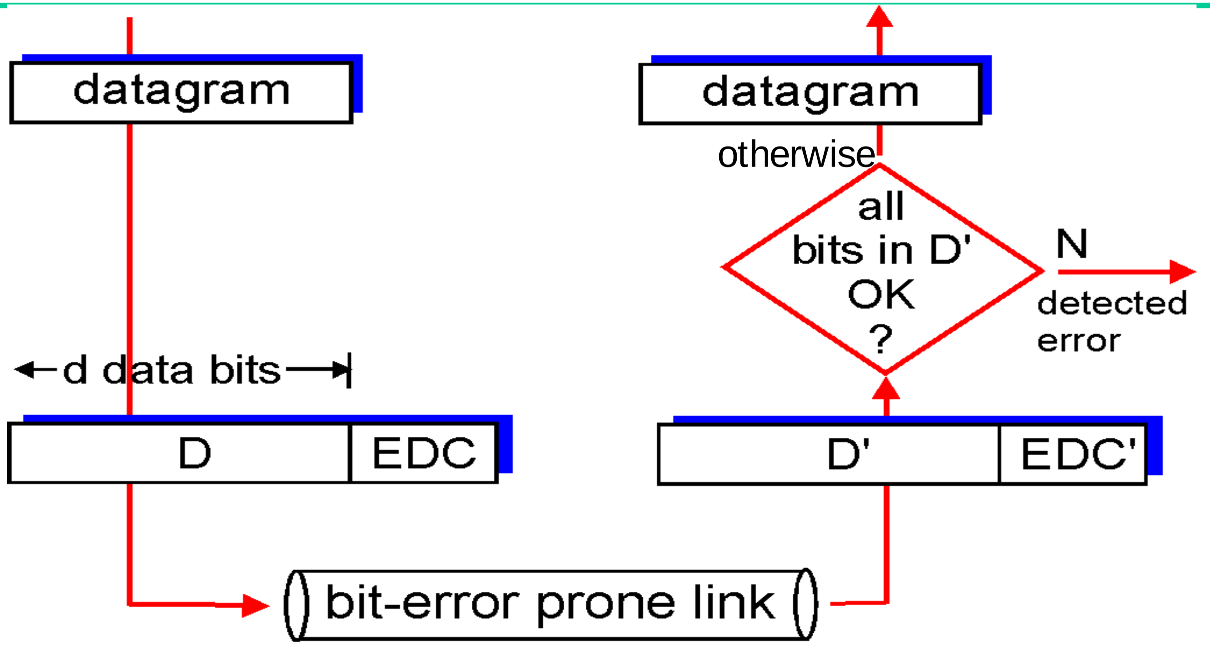
Error-Correcting Codes (not in the scope of this course)

- Hamming codes.
- Binary convolutional codes.
- Reed-Solomon codes.
- Low-Density Parity Check codes

Error detection

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

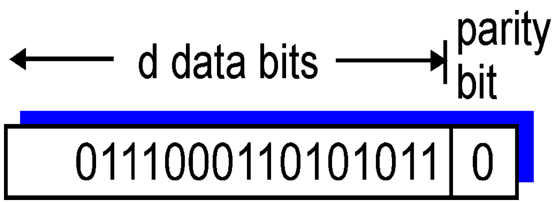
- Error detection not 100% reliable!
  - protocol may miss some errors, but rarely
  - larger EDC field yields better detection and correction



Parity checking

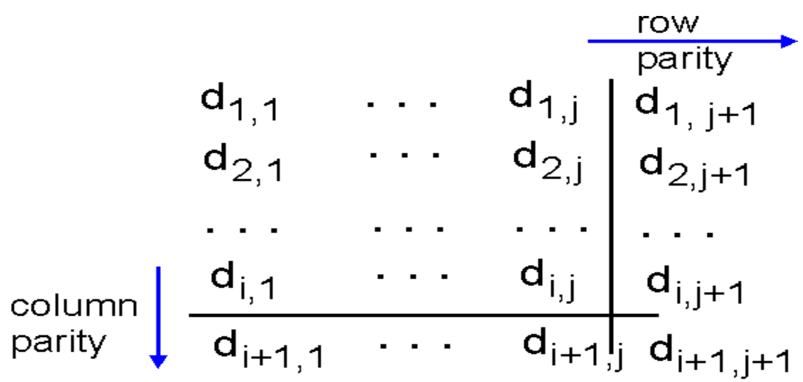
single bit parity:

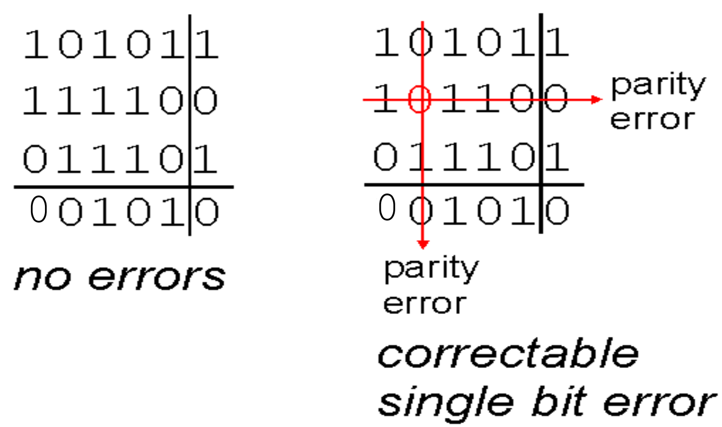
- *d*etect single bit errors



two-dimensional bit parity:

- detect and correct single bit errors





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 → 0110011?
  0110001 → 0110001?
- Performance:
  - Detects
    - single-bit errors
    - multiple-bit or burst errors only if the total number of errors is odd

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

Example

- Suppose the information content of a packet is the bit pattern 1110 0110 1001 1101 and an even parity scheme is being used. What would the value of the field containing the parity bits be for the case of a two-dimensional parity scheme?

1 1 1 0 ?

0 1 1 0 ?

1 0 0 1 ?

1 1 0 1 ?

? ? ? ? ?

Example

- Suppose the information content of a packet is the bit pattern 1110 0110 1001 1101 and an even parity scheme is being used. What would the value of the field containing the parity bits be for the case of a two-dimensional parity scheme?

1 1 1 0 1

0 1 1 0 0

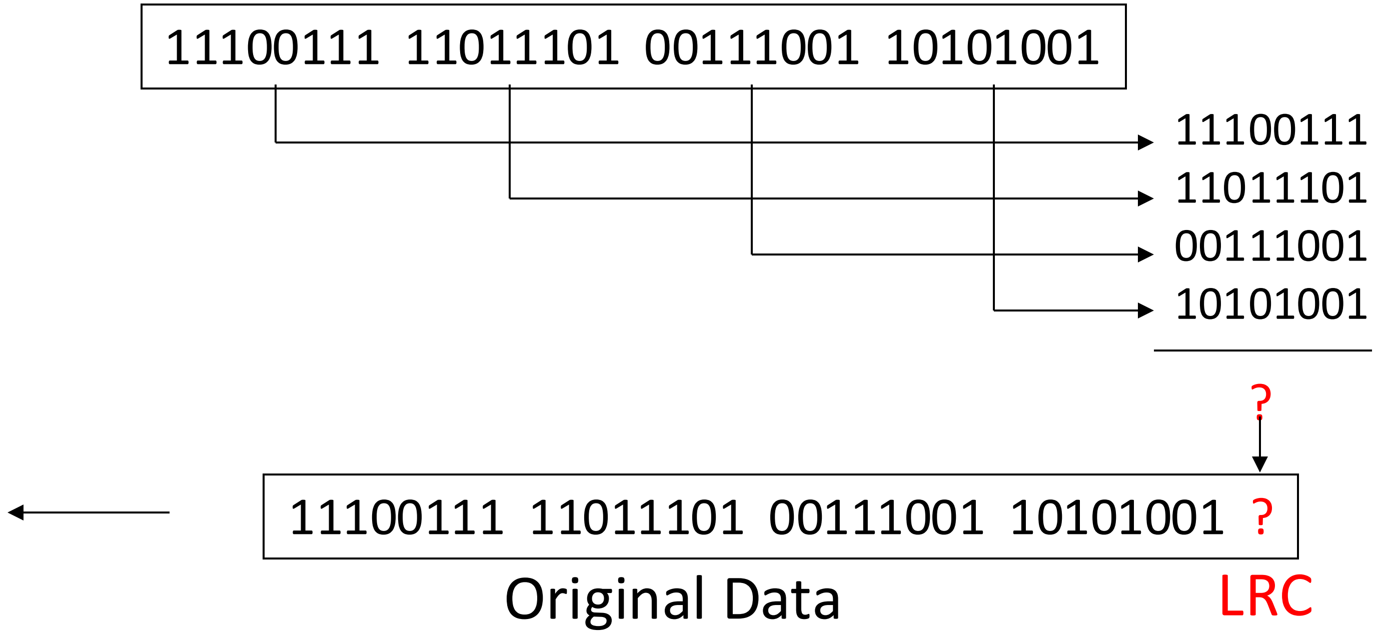
1 0 0 1 0

1 1 0 1 1

1 1 0 0 0

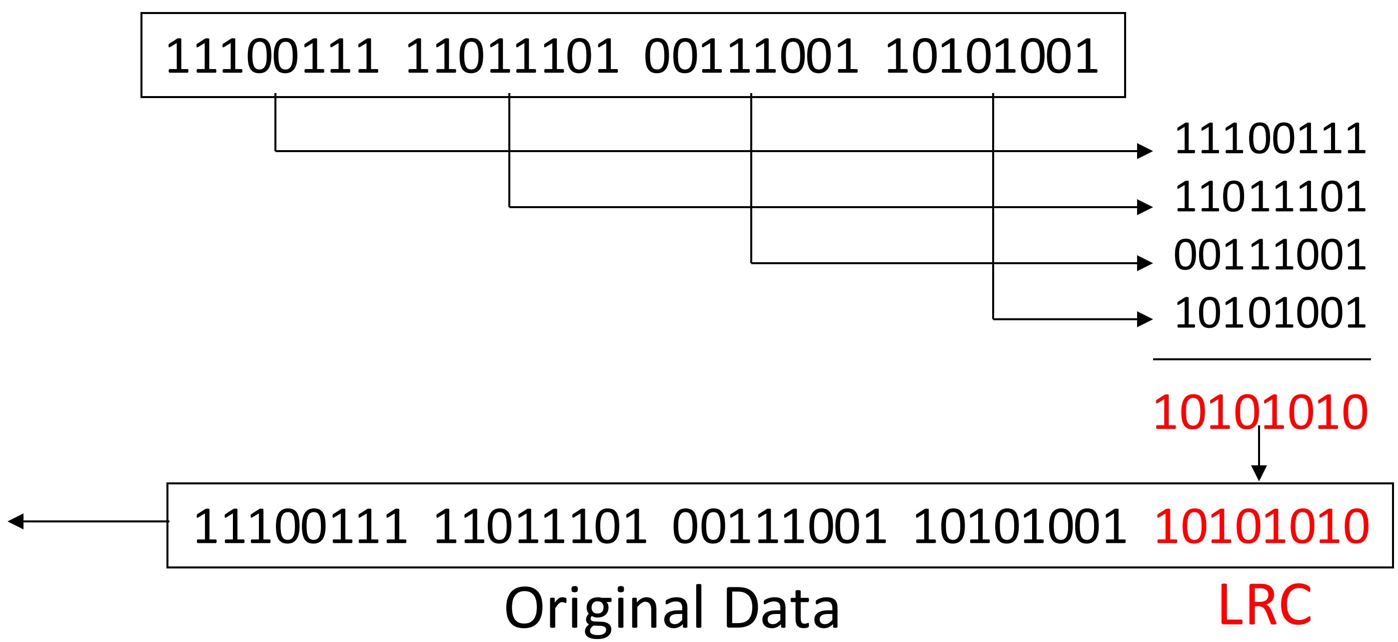
Longitudinal redundancy check (LRC)

- 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



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



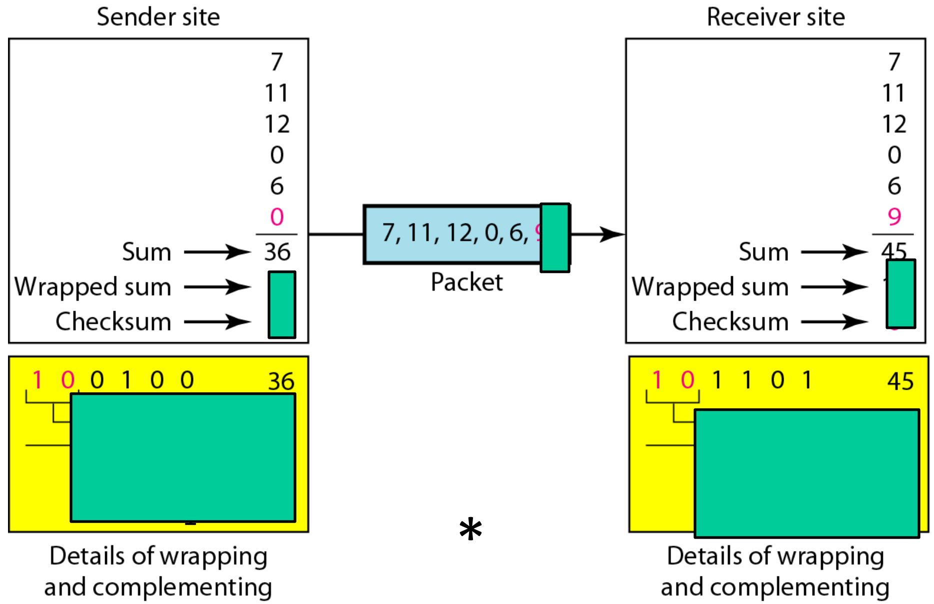
Checksum

Idea:

- Message is five 4-bit numbers: 7, 11, 12, 0, 6
- The sender sends message+ redundancy,
  - numbers + sum: 7, 11, 12, 0, 6, 36
- The receiver adds five numbers and compares the result with the sum
- Easier if negative (complement) of the sum (i.e. -36) is sent
  - Only 4-bits can be used
- 1's complement arithmetic , represent unsigned numbers
  - n-bits is used to represent numbers $2^n - 1$
  - If the number has more than n bits, the extra left most bits to be added to the n rightmost bits (wrapping)
  - In 1's complement, the negative number is represented by inverting all bits. Same as subtracting the number from $2^n - 1$

Checksum

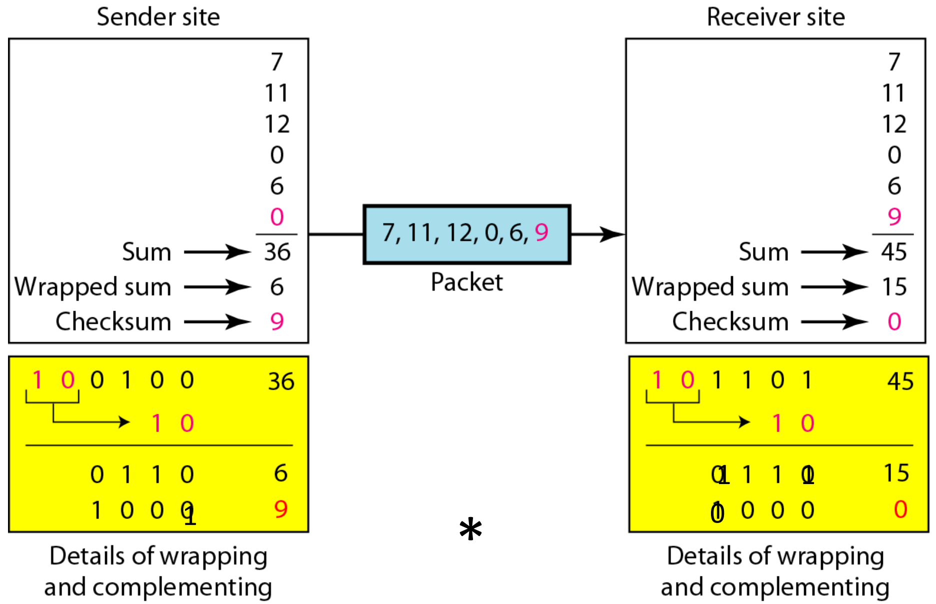
- $1^{'s}$ complement of number 21 using 4-bits → 10101 (5 bits),
- So wrap the leftmost bit and add it to 4 rightmost bits: 0101+1 = 0110 (6)
- - 6 in 1's complement is 1001 → 9, so complement of 6 is 9
  - Or subtract 6 from $2^n - 1$ (15)
- Example: at the sender: 36 → 100100 in 4-bits = 0100+10 =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

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

Cyclic redundancy check (CRC)

more powerful error-detection coding

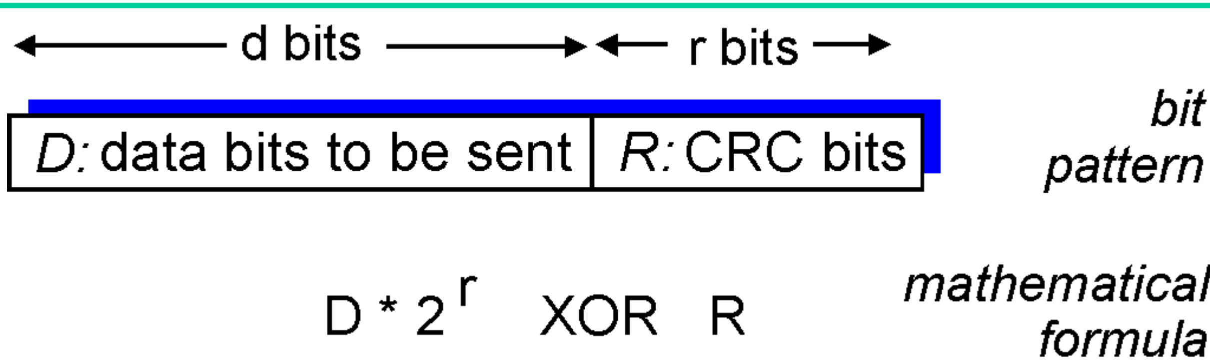
view data bits, D, as a binary number

choose r+1 bit pattern (generator), G

goal: choose r CRC bits, R, such that

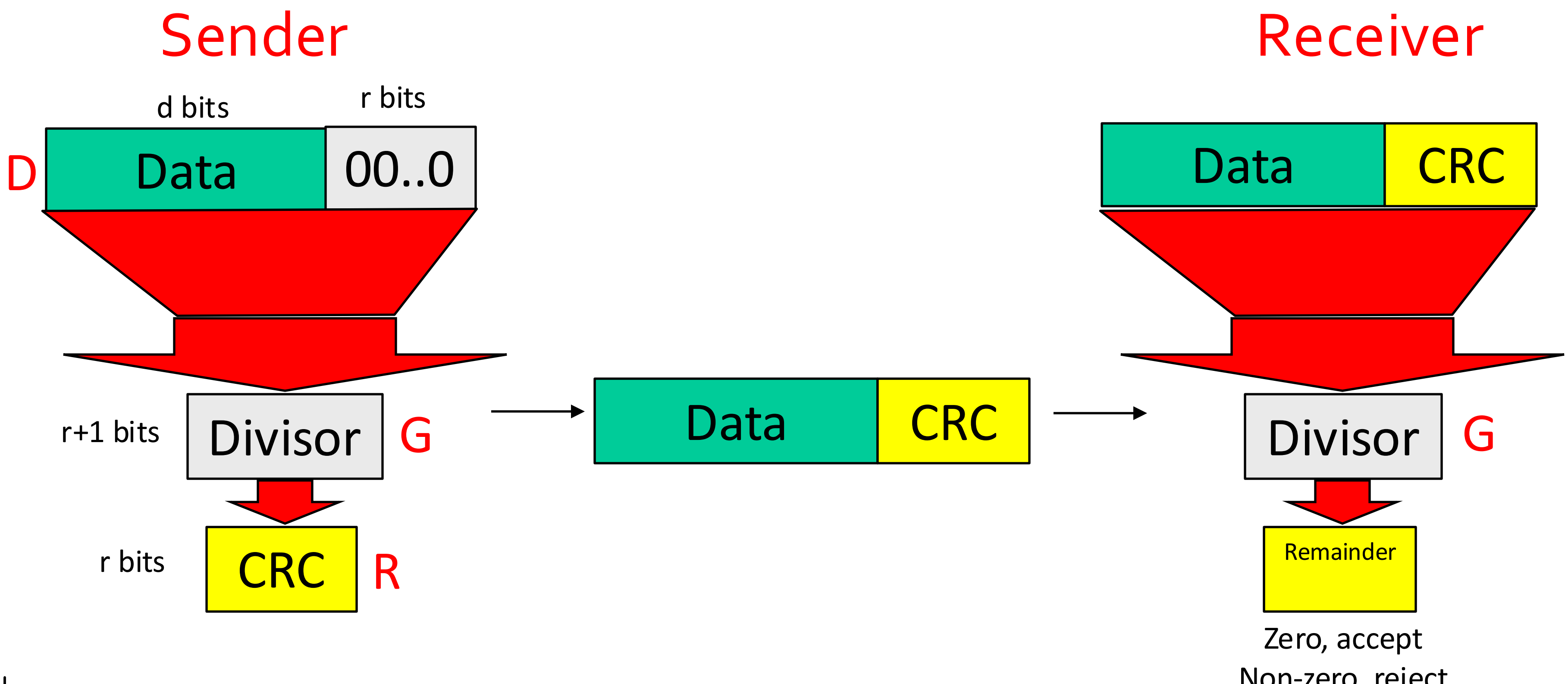
- <D,R> exactly divisible by G (modulo 2)
- receiver knows G, divides <D,R> by G. If non-zero remainder: error detected!
- can detect all burst errors less than r+1 bits

widely used in practice (Ethernet, 802.11 WiFi)



Cyclic Redundancy Check

Rather than addition, binary division is used → Finite Algebra Theory (Galois Fields)



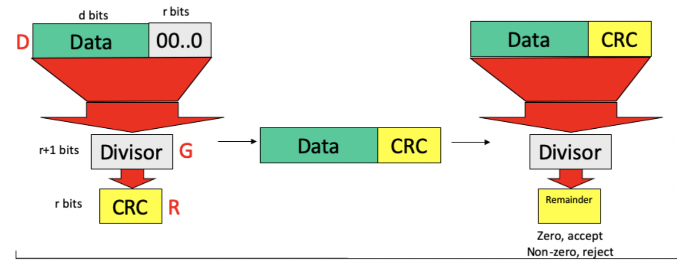
$D.2^r$ XOR R

- CRC calculations are done in modulo 2 arithmetic
  - without carries in addition and without borrows in subtraction.
  - Addition and subtraction are identical, and both are equivalent to the bitwise exclusive or, XOR, of the operands
  - e.g. 11001 XOR 01011 = 10010
  - 11001 + 01011 = 10010
  - 11001 - 01011 = ?
  - 10010 XOR 01011 = 11001

→ A XOR B = C, then A = C XOR B

- $D.2^r$ → shift left D by r bits → add r 0's to the right of D
- e.g. $D.2^5$ → shift left D by 5 bits

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:

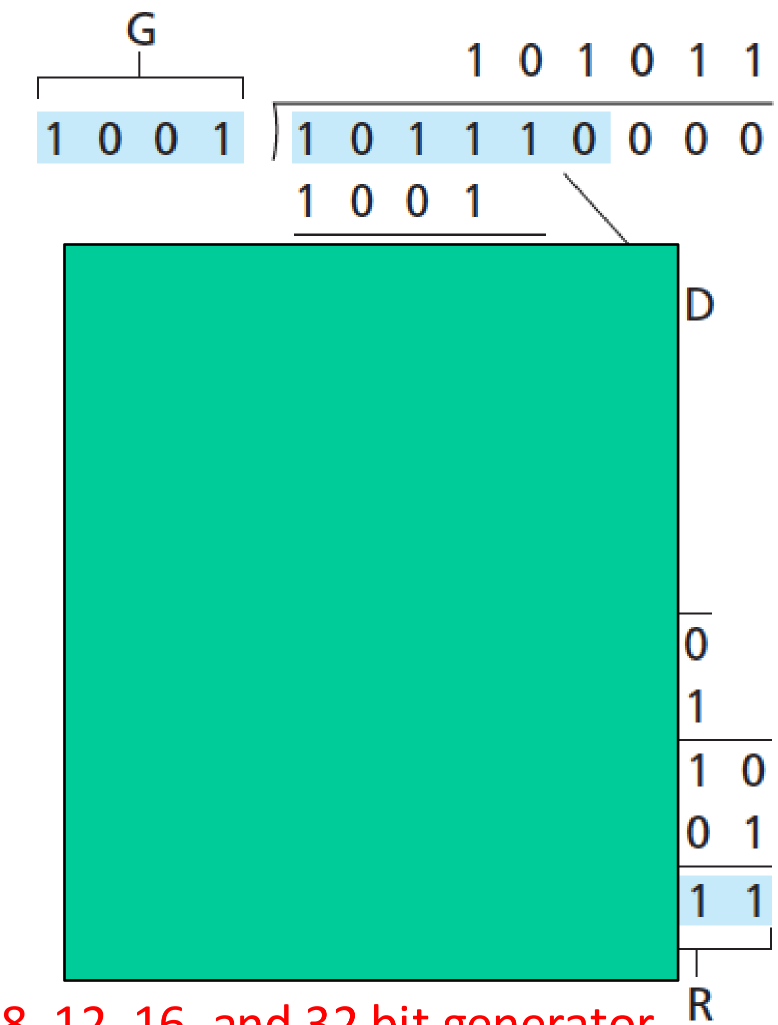
$$R = remainder\left[\frac{D \cdot 2^r}{G}\right]$$

How?

e.g D = 101110, G = 1001 → r = 3

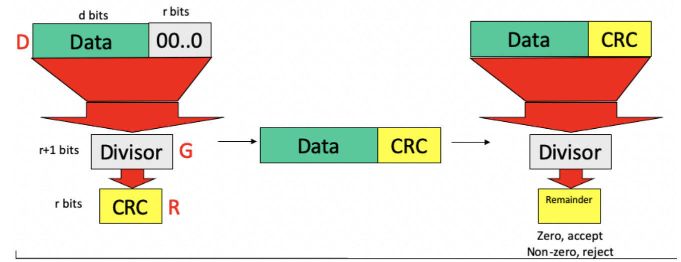
Check the leading left bit is 0 or 1.

- 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

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:

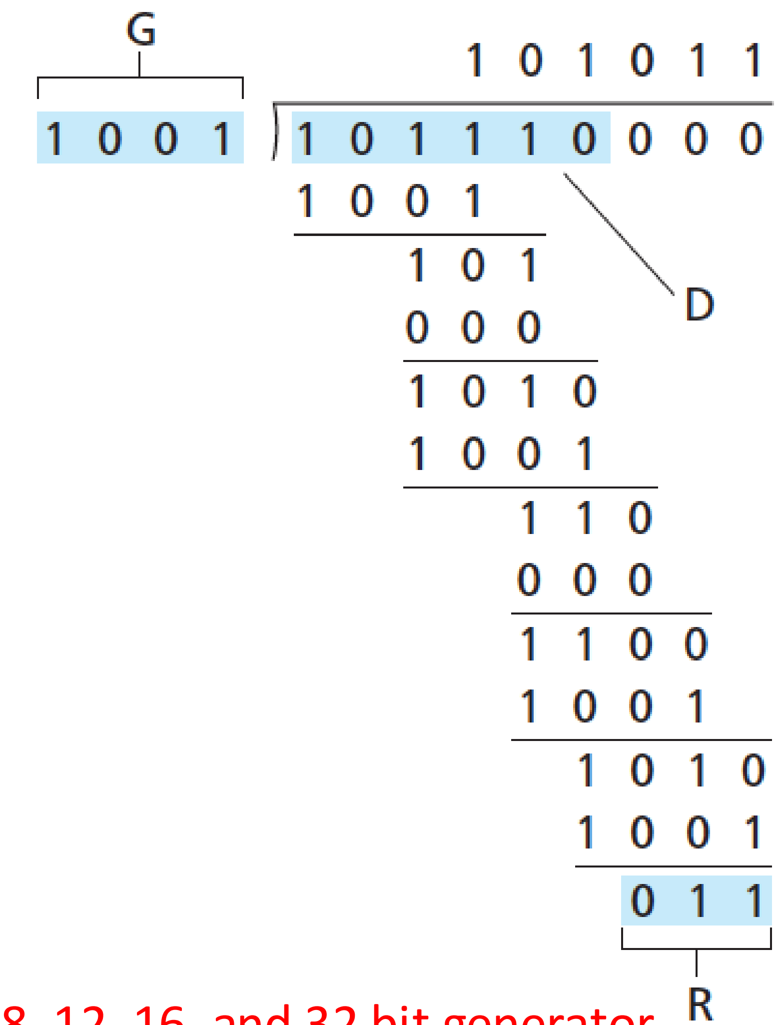
$$R = remainder\left[\frac{D \cdot 2^r}{G}\right]$$

How?

e.g D = 101110, G = 1001 → r = 3

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

Summary

Today:

- Data link layer
- Error detection and error correction
  - Parity
  - Checksum
  - Cyclic Redundancy Check

Canvas discussion:

- Reflection
- Exit ticket

Next time:

- read 6.3 of KR (Multiple Access Links and Protocols)
- follow on Canvas! material and announcements

Any questions?