Computer Networks

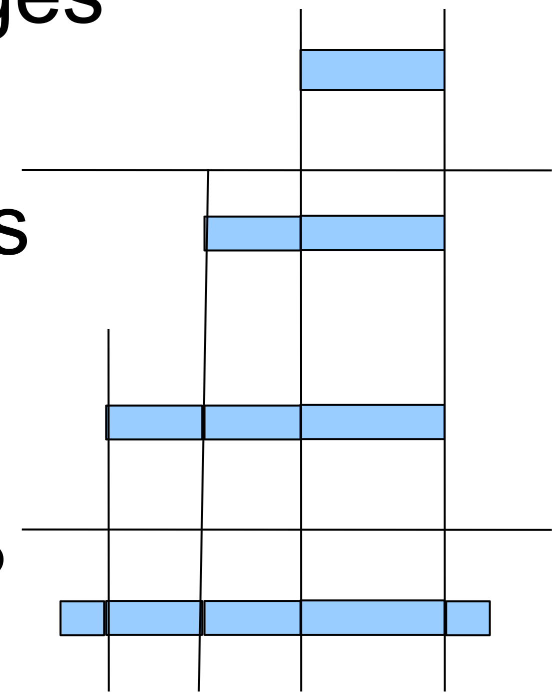
Link Layer

Jianping Pan
Fall 2022


NASA
ARTEMIS
STUDENT CHALLENGES

Review

- Application layer: messages
  - HTTP, DNS
- Transport layer: segments
  - TCP, UDP
- Network layer: packets
  - IP, ICMP; RIP, OSPF, BGP
- Link layer: frames



Q: end-to-end, hop-by-hop?

Link layer services

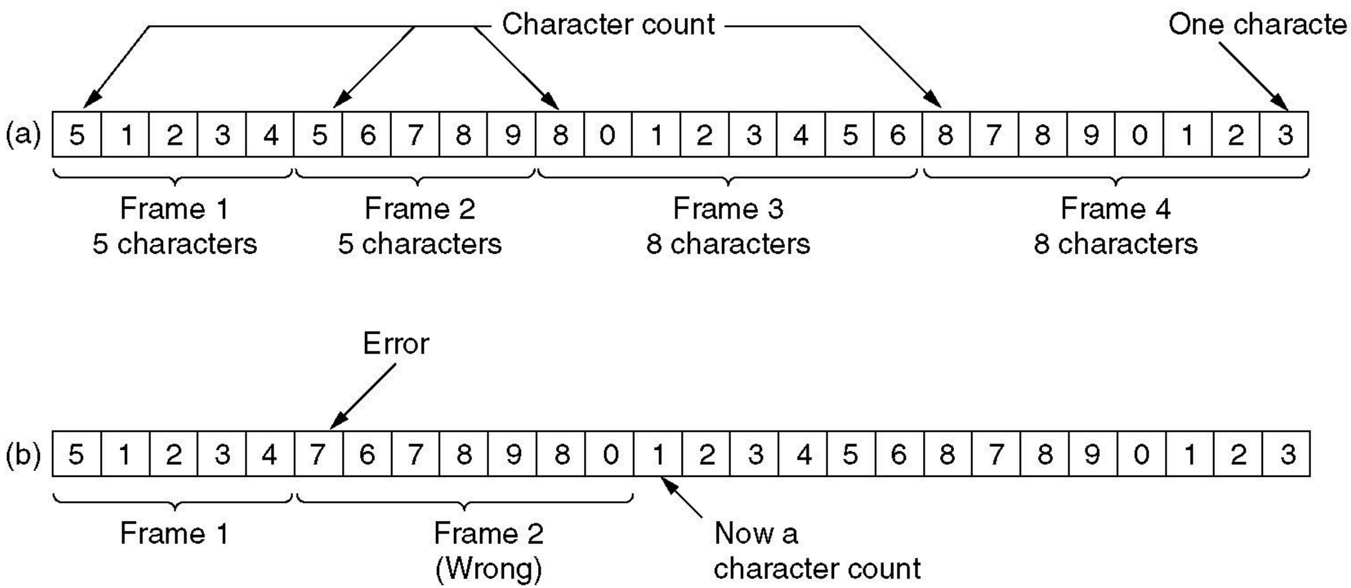
- Services provided by physical layer
  - bit delivery (recall: access networks)
    - hertz, baud, symbol-per-second, bit-per-second
- Services provided to network layer
  - frame control: framing
  - error control: how to deal with bit errors
  - flow control: fast sender vs slow receiver
  - medium access control (with shared medium)

Today's topics

Q: Nyquist's Limit, Shannon's Limit

Frame control

- Character count
  - count error, and error propagation

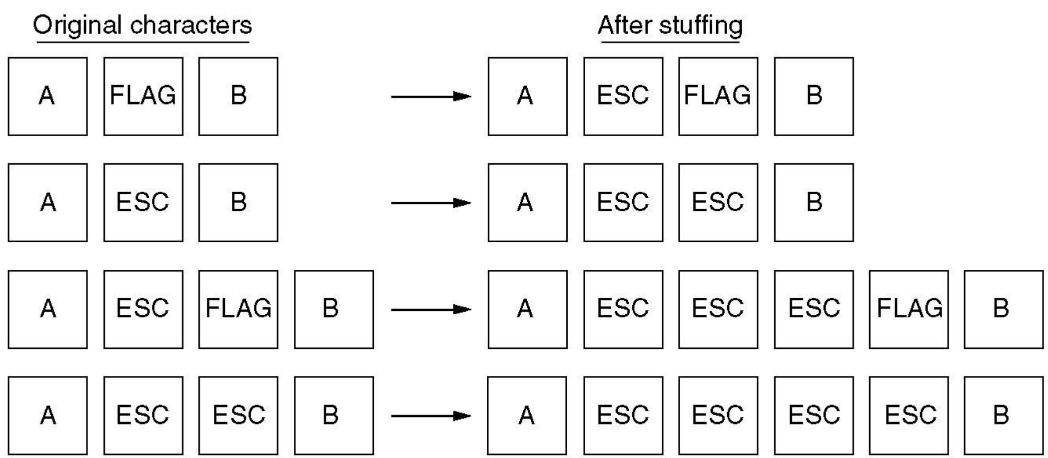


Q: Pascal vs C strings

Byte-oriented framing

- Byte stuffing

(a)



(b)

Q: error propagation?

Bit-oriented framing

- Flag: 01111110
  - data transparency: bit stuffing
    - sender: insert a 0 after 5 1's
    - receiver: remove a 0 after 5 1's

(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

Error control

- Hamming distance of codeword *a* and *b*
  - number of *pairwisely* different bits
    - number of bit flips needed to turn *a* to *b*

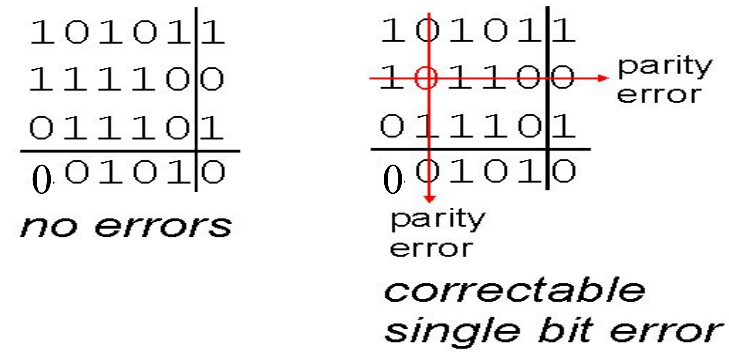
```
     01010101
XOR) 00100100
     --------
     01110001
            4
```

- Hamming distance of codeword set $\{a_i\}$
  - minimal distance btw a_i and a_j, where i != j
- *e* bit errors
  - to detect: minimal Hamming distance *e*+1
  - to correct: minimal Hamming distance 2*e*+1

* Hamming, "You and Your Research" (June 6, 1995) https://www.youtube.com/watch?v=a1zDuOPkMSw

Parity check

- Parity bit
  - even or odd parity
  - i.e., the number of 1's
  - e.g., 10101; check bit: 1 (even)
  - Q: Hamming distance?
  - detect 1-bit error
- 2-d parity
  - correct 1-bit error



Q: how to count 1's *quickly*?

Hamming code

- Hamming code
  - check bits
    - at bit 1, 2, 4, 8
  - data bits
    - at bit 3, 5, 6, 7, 9, 10, 11
  - e.g, 1001000
  - correct 1-bit error
- Hamming code block
  - correct up to block length

Char.	ASCII	Check bits
H	1001000	00110010000
a	1100001	10111001001
m	1101101	11101010101
m	1101101	11101010101
i	1101001	01101011001
n	1101110	01101010110
g	1100111	01111001111
	0100000	10011000000
c	1100011	11111000011
o	1101111	10101011111
d	1100100	11111001100
e	1100101	00111000101

Order of bit transmission

Review: Internet checksum

- Checksum: widely used in upper layers
  - e.g., TCP checksum with pseudo header
  - optional UDP checksum with pseudo header
  - IP header checksum
- One's complement of one's complement sum
  - checksum generation
  - checksum verification
- When does checksum fail?

Checksum generation	Checksum verification
AA BB	AA BB
CC DD	CC DD
EE 00	EE 00
+) 00 00	+) 9A 65
265 98	2FF FD
65 9A	FF FF
9A 65	00 00

Cyclic Redundancy Check

- CRC: widely used in lower layers
  - e.g., IEEE 802.3 CRC-32-Ethernet
  - ITU-T X.25 CRC-16-CCITT
- Polynomial representation
  - message: M(x); generator: G(x) of order *r*
  - remainder: $R(x)=M(x)*2^r \% G(x)$
  - CRC generation: $T(x) = M(x)*2^r$ XOR $R(x)$
    - i.e., T(X) is G(x) divisible
  - error: E(x) detected if not G(x) divisible

Q: trailer?

CRC Example

Want:

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

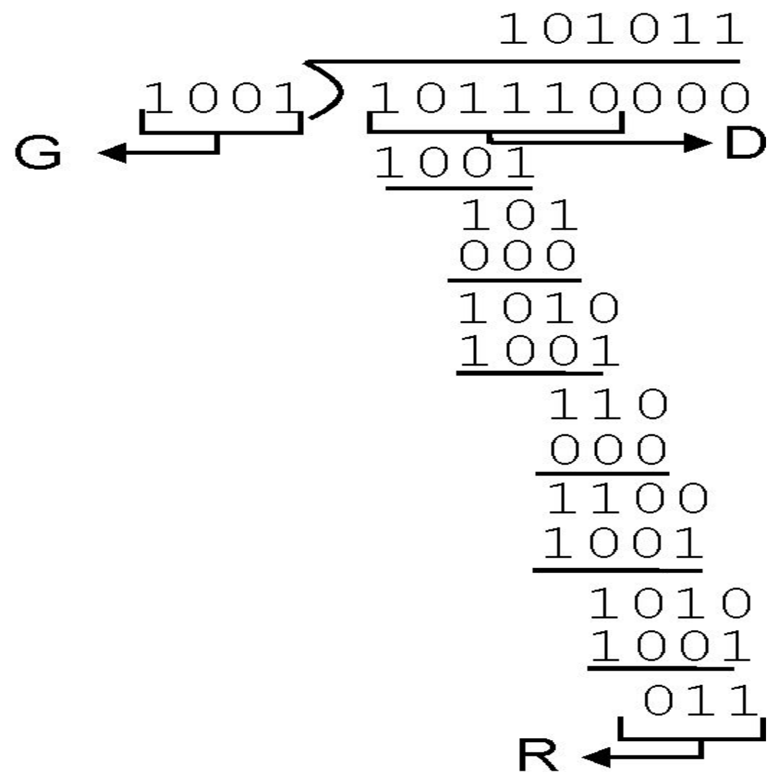
equivalently:

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

equivalently:

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

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



Q: parity check's G(x)?

Error recovery

- Positive acknowledgment
  - cumulative acknowledgment
    - acknowledge packet x: acknowledge packets 1..x
    - when timeout, go-back-N
  - selective acknowledgment
    - acknowledge packet x: packet x is received OK
    - when timeout, selective repeat
- Negative acknowledgment
  - report: x is corrupted or *missing*

This lecture

- Link layer
  - framing
  - error control
    - error detecting, error correcting, error recovery
- Explore further
  - Information and Coding Theory
    - 1850s-1940s: check digit; 1940s-1960s: checksum
    - 1960s: Reed-Solomon; 1970s: LDPC codes
    - 1980s: Turbo codes; 1990s: Space-time code
    - 2000s: Polar code

Next lectures

- Flow control and LLC protocols
  - sliding window (1-bit, GBN, SR)
  - read K&R4: Computer Networking
    - Chapter 5 (except 5.7 and 5.8)
    - newer editions: Chapter 6 (Link Layer)