# HW #3: Networking Questions

Spring 2020

1. Bit Stuffing.

a. A bit string, 10001111110100011111011, needs to be transmitted at the data link layer. What is the string transmitted across the Link after bit stuffing by the sender? Assume the same start/end flags as the ones used in class.

The string after bit stuffing:

01111110 1000 111110 101000 111110011 01111110

b. A frame is received by the data link layer, which was transmitted using bit stuffing: 01111110111110110001111101101111110. What is the bit string that the link layer passes up the stack to the network layer after bit de-stuffing?

After bit de-stuffing:

1111111 000 1111111

2. Hamming Code.

a. Encode the message 10011011 to send.

Hamming Table

| | | | 1 | | 0 | 0 | 1 | | 1 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | b1 | b2 | b3 | b4 | b5 | b6 | b7 | b8 | b9 | b10 | b11 | b12 |
| | P1 | P2 | d1 | P4 | d2 | d3 | d4 | P8 | d5 | d6 | d7 | d8 |
| C1 | X | | X | | X | | X | | X | | X | |
| C2 | | X | X | | | X | X | | | X | X | |
| C3 | | | | X | X | X | X | | | | | X |
| C4 | | | | | | | | X | X | X | X | X |
| | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | $\boxed{1}$ | 0 | 1 | 1 |
| | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | $\boxed{1}$ | |

Since the bits in each row needs to be even parity,

p1 d1 d2 d4 d5 d7
<u>0</u> 1 0 1 1 1

p2 d1 d3 d4 d6 d7
<u>1</u> 1 0 1 0 1

P4 d2 d3 d4 d8
<u>0</u> 0 0 1 1

P8 d5 d6 d7 d8
<u>1</u> 1 0 1 1

The encoded message will be:

011000111011

b. What can be said about the correctness of the following received messages ( Hint: Check for Hamming Code correctness using parity)?

i. 111000101011

ii. 01110011011

i. By using the Hamming Table, we can recompute the parity.

C1 = 1 C2 = 0 C3 = 0 C4 = 1

So, i message's bit 9 is flipped. We can correct b9 = 1 to b9 = 0.

ii. C1 = 1 C2 = 1 C3 = 0 C4 = 1

These shows that bit 11 is flipped. It's original value should be 0.

3. CRC Code. Assume the C(x) = $x^4 + x^2 + 1$.

a. Encode the message 10110 with CRC.

b. What can be said about the correctness of the following received messages?

i. 110101110

ii. 110101100

a. C is 10101   The degree k of C(x) is 4

m = 10110   n_(shifted m) = 101100000

Compute the division:   r = 1111

```
        10011
10101 ) 101100000
        10101
          11000
          10101
           11010
           10101
            1111
```

CRC encoded message:

101101111

b. i.

```
        111
10101 ) 110101110
        10101
         11111
         10101
          10101
          10101
             10 → remainder
```

The remainder is not zero.

So, the message is not correct

ii.

```
        111
10101 ) 110101100
        10101
         11111
         10101
          10101
          10101
              0
```

The remainder is 0.

So, this message is correct.

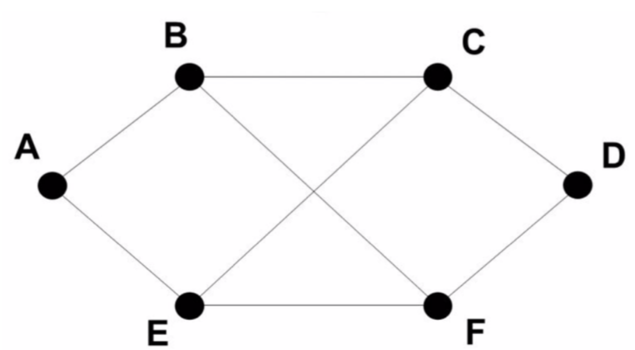
4. **Distance Vector Routing. Distance Vector Routing.** Consider the subnet shown below. Distance vector routing is used, and the following distance vectors have just come in to router C: **B**: (6, 0, 8, 10, 5, 5); from **D**: (4, 9, 7, 0, 8, 6); and from **E**: (7, 7, 4, 8, 0, 5). The measured distances/costs from C to **B**, **D**, and **E** are 5, 5, and 4, respectively. What will C's new routing table be after this update? Show both the outgoing router to use and the cost.

A B C D E F

B: (11, 5, 13, 15, 10, 10)

D: (9, 14, 12, 5, 13, 11)

E: (11, 11, 8, 12, 4, 9)




Routing Table Format:

| Destination | Cost | Next Hop |
|---|---|---|
| A | 9 | D |
| B | 5 | B |
| C | 0 | — |
| D | 5 | D |
| E | 4 | E |
| F | 9 | E |

5. TCP Sequence Numbers. To get around the problem of sequence numbers wrapping around while old TCP packets still exist, TCP could use 64-bit sequence numbers instead of 32 bits. However, theoretically, an optical fiber can run at 100 Terabits per second. What maximum packet lifetime would be required to prevent sequence number wrap-around even with 64-bit sequence numbers? Assume that each byte of a packet has its own sequence number (as TCP does).

Total Possible Sequence Numbers are $2^{64}$

The Bandwith is 100 Terabits/second $= 12.5$ TB/s $= 12.5 \times 2^{40}$ bytes/s.

So, the wrap around time for 64-bit sequence number $= 2^{64} / (12.5 \times 2^{40}\ B/s) = 1.28 \times 2^{20}\ s \approx 15.53$ days

6. DNS. Using an online whois lookup service like whois.net, look up duke.edu. On what date was the domain registered? When does it expire? What are the DNS servers for this domain? Include a screenshot of your source.

**duke.edu is already registered***

Domain Name: DUKE.EDU
Registry Domain ID: 5059_DOMAIN_EDU-VRSN
Registrar WHOIS Server: whois.educause.net
Registrar URL: http://www.educause.edu/edudomain
Updated Date: 2018-06-08T13:57:29Z
Creation Date: 1986-06-02T04:00:00Z
Registry Expiry Date: 2021-07-31T11:59:59Z
Registrar: Educause
Registrar IANA ID: 365
Registrar Abuse Contact Email:
Registrar Abuse Contact Phone:
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Name Server: DNS-AUTH-01.OIT.DUKE.EDU
Name Server: DNS-AUTH-02.OIT.DUKE.EDU
Name Server: DNS-NC1-01.OIT.DUKE.EDU
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2018-06-13T13:34:24Z <<<

Creation Date: 1986-06-02T 04:00:00Z
Expire Date: 2021-07-31T 11:59:59Z
Name Server: DNS-AUTH-01.OIT.DUKE.EDU
DNS-AUTH-02.OIT.DUKE.EDU
DNS-NC1-01.OIT.DUKE.EDU

7. Internet Services. Using netcat (the ‘nc’ command) in a terminal, manually display the following URL to the console.
http://rabihyounes.com/awesome.txt

```
wz125@vcm-12917:~$ printf "GET http://rabihyounes.com/awesome.txt\r\n HTTP/1.1" Host: rabihyounes.com | nc rabihyounes.com 80
```