HW #3: Networking Questions

- Student Name: Yifan Men
- Student NetID: ym129

Bit Stuffing.

(a) A bit string, 100111111001011111100011, needs to be transmitted at the data link layer. What is the string transmitted across the Link after bit stuffing by the sender, assuming the bit stuffing scheme shown in the lecture slides?

My answer:
01111110 10011111**0**1001011111**0**00011 01111110

(b) A frame is received by the data link layer, which was transmitted using bit stuffing: 0111111000111110110011111001101111110. What is the bit string that the link layer passes up the stack to the network layer after bit de-stuffing, assuming the bit stuffing scheme shown in the lecture slides?

My answer:
~~01111110~~ 0011111~~0~~110011111~~0~~011 ~~01111110~~ $=>$00111111100111111011

Link Layer Protocols.

A channel has a bit rate of 4 kilobits per second and a propagation delay of 20 milliseconds. For what range of frame sizes does stop-and-wait give a link utilization efficiency of at least 50%?

My answer:

Bit rate = 4 kbps
Propagation delay = 20 ms

$$efficiency = \frac{transmission\_time\_of\_packet}{transmission\_time\_of\_packet + 2 * propagation\_delay}$$

$$0.5 = \frac{xsec}{xsec + 2 * 20 * 10^{-3} sec} => x = 40 * 10^{-3} sec$$

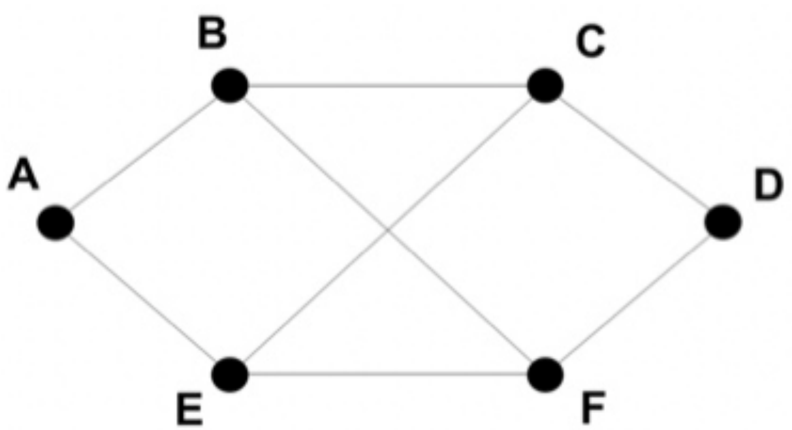
$$\frac{minimum\_frame\_size}{bit\_rate} = 40 * 10^{-3} sec$$

$$minimum\_frame\_size = 40 * 10^{-3} sec \cdot 4 * 10^3 bits/sec = 160 bits$$

Therefore, when the frame sizes is greater than $160 bits$, stop–and–wait give a link utilization efficiency of at least 50%.

Distance Vector Routing.

Consider the subnet shown below. Distance vector routing is used, and the following vectors (showing the cost from each node) have just come in to router C: from B: (5, 0, 9, 12, 6, 2); from D: (16, 12, 6, 0, 9, 10); and from E: (7, 6, 3, 9, 0, 4). The measured delays from C to B, D, and E are 6, 6, and 3, respectively. What will C's new routing table be after this update? Show both the outgoing line to use and the expected delay.



My answer:

Destination	Cost	Next Hop
A	10	E
B	6	B
C	0	/

D	6	D
E	3	E
F	7	E

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 75 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).

My answer:

The size of data is: $2^{64} bytes$, which is $2^{64} * 8 = 2^{67} bits$

The optical fiber run at $75Tbps$, that is $75 * 2^{40} bits/sec$

The time to transmit the data would spend

$$\frac{2^{67} bits}{75 * 2^{30} bits/sec * 86400 sec/day} = 20.7 days$$

The maximum packet lifetime would be $20.7 days$.

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.

My answer:

- Creation Date: 1986-06-02T04:00:00Z
- Registry Expiry Date: 2021-07-31T11:59:59Z
- Name Server
    - DNS-AUTH-01.OIT.DUKE.EDU

- DNS-AUTH-02.OIT.DUKE.EDU
- DNS-NC1-01.OIT.DUKE.EDU

WHOis.net

WHOIS LOOKUP

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

Internet Services.

Using netcat (the ‘nc’ command) in a terminal, manually display the following URL to the console.

http://rabihyounes.com/awesome.txt

My answer:

```
nc rabihyounes.com 80
```

```
GET /awesome.txt HTTP/1.1
Host: rabihyounes.com
```

Output: