Solver: Chen Zhe

Email Address: chen0892@e.ntu.edu.sg

I do hope my solutions can be provide you with a better understanding as I do have found quite a number of inaccuracies in previous year's solutions.

1. Apply Yonggang's CRACK framework whenever possible

(a)

Context: it's a network reliability problem with only serial links

Framework: link success probability of n serial links is calculated by $(1-b)^n$ where b is the probability of failure of each individual link

Application: the link success probability should be greater than 99%. Each link has a failure possibility of 10^{-3}. As such: $(1-10^{-3})^n \geq 99\%$

Calculation:

apply $\log_{10}$ on both side of the inequation.

$n*\log_{10}(1-10^{-3}) \geq \log_{10}(0.99)$

note that $1-10^{-3} < 1 \rightarrow \log_{10}(1-10^{-3}) < 0 \rightarrow$ the direction of inequation is reversed when doing the division.

$n \leq \log_{10}(0.99) / \log_{10}(0.999)$

$n \leq \log_{10}(0.99) / \log_{10}(0.999)$

$n \leq 10.04$

round down

$n \leq 10$

Check: 10 is a greater than 0, and is a reasonable number of links.

Therefore the maximum number of sequential links allowed is 10 to provide 99% reliability.

(b)

(i) Context: it's a MARP problem

Framework: the throughput of MARP is S=u/(u+v/S_r) where u is the transmission time of the data frame, v is the transmission time of the reservation frame and S_r is the link utilization rate of the MAC protocol used in the reservation phase

Application: v = 5 bytes / 4 Mega bits per second = 0.01 second, u = 125 bytes / 4Mbps = 0.25 second, S_r = 0.5

wrong!

Calculation: S = 0.25/(0.25+ 0.01 / 0.5) = 0.9259

Check: the utilization is smaller than 1, which is reasonable

(ii) Context: it's a MARP + slotted Aloha problem

Framework: the throughput of MARP is S=u/(u+v/S_r) where S_r = Ge^{-G} is the link utilization rate of slotted Aloha protocol used in the reservation phase

Application: v = 0.01 second, u = 0.25 second, G = np

As the wifi in student lounge is used by over 200 students, it can be assumed that n = 200.

Calculation: the maximum S_r is reached when G = 1, and S_r (max) = e^{-1}.

As such, the optimal probability of transmission is p = G/n = 1/200

S = 0.25/(0.25+ 0.01 * e) = 0.9019

Check: the utilization is smaller than 1, which is reasonable

(c)

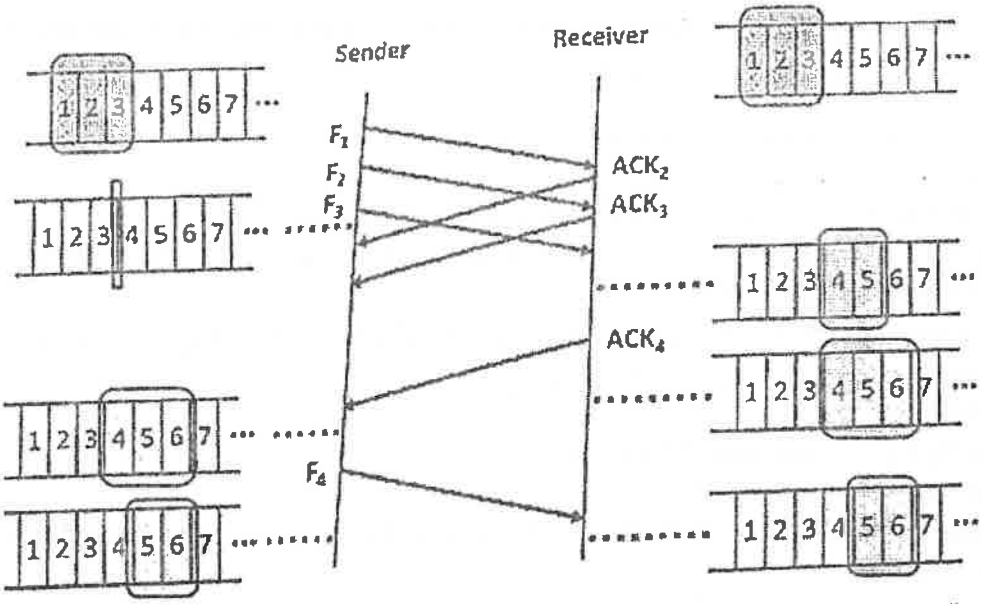


Figure Q1

2. (a)

(i) Context: it's an error control with selective-reject ARQ problem

Framework: the link utilization of a selective-reject ARQ scheme is

$$U = \begin{cases} 1-P, & W \geq 1+2a \\ \dfrac{W(1-P)}{(1+2a)}, & W < 1+2a \end{cases}$$

, where *W* is the window size, *P* is the probability of error and *a* is

$$a = \frac{T_{prop}}{T_{frame}}$$

Application: P = 0.2, W = 12, a = (15km * 1ms/km) / (200bits/100bits per millisecond) = 7.5

Calculation:

since W = 12 ≤ 16 = 1 + 2a, the second formula applies

U = 12*(1-0.2) / 16 = 0.6

Check: 0.6 is smaller than 1 so it's a reasonable number of link utilization

(ii) Context: it's a link utilization maximization problem

Framework: the link utilization of a selective-reject ARQ is stated in (i) and the maximum link utilization is reached when W≥1+2a. the maximum windows size possible for *k* bits frame sequencing in selective-reject ARQ is 2^{k-1}.

Application and calculation:

as calculated in (i), a = 16, therefore the minimum windows size W = 16.

Therefore, 16 = 2^{k-1}, $\log_2(16) = (k-1) * \log_2(2)$, 4 = k-1, k = 5

Check: both W and k are integers. (if W is not integer, round up and then calculate k)

(b)

(i) Context: it's a queue with feedback problem

Framework: the effective load is the system utilization: $\rho = \frac{\lambda}{\mu}$: system utilization, where λ is the arrival rate and μ is the service rate. Expected total delay is $T = \frac{1}{\mu-\lambda}$: average delay.

Application: λ = 320 frames/s without accounting for feedback. μ = 1000kbps/200bits per packet = 500 frames/s

Calculation:

the real arrival rate λ' = λ+ λ'*P → λ' = 320 / (1-0.2) = 400 frames/s

λ < μ holds thus the system is stable, therefore all formulas applies

effective load = 400/500 = 0.8

expected total delay = 1/(500-400) = 0.01s = 10ms

check: effective load is smaller than 1, expected total delay is a positive number

(ii) Context: it's a MM1 queue packet delay components problem

Framework:

Delay Components for Packet:
- Average residual service time: ρ/μ
- Average waiting time for early arrival: $(N-\rho)/\mu$
- Average service time: $1/\mu$

$N = \frac{\rho}{1-\rho}$: average queue occupancy

Application: from (i): ρ = 0.8, μ = 500, N = 0.8/(1-0.8) = 4

Calculation:

Average residual service time = 0.8/500 = 1.6ms

Average waiting time for early arrival = (4-0.8)/500 = 6.4ms

Average service time = 1/500 = 2ms

Check: total delay is 1.6 + 6.4 + 2 = 10ms and agrees with the result from (i)

3. (a)

	IP address		TCP port number	
Packet	Source	Destination	Source	Destination
1	192.168.1.68	155.69.8.9	50000	80
2	13.5.6.7	**155.69.8.9**	55000	**80**
3	155.69.8.9	13.5.6.7	58000	**55000**
4	**192.168.1.65**	192.168.1.68	58000	**50000**

(b)

(i) The first 16 bits (193.53) matches only the 3rd entry of the routing table. Bits 17-23 of the IP address are: 0011100, bits 17-23 of 3rd entry of the routing table is 0010100, they do not match. Therefore, the default entry is to be used and it will be routed to router 2.

(ii) The first 16 bits of the IP address (135.46) matches the 1st and 2nd entry of the routing table. Bits 17-22 of the IP address are: 001110. Bits 17-22 of 1st routing table entry are: 001110, bits 17-22 of the 2nd routing table entry are: 001111, therefore the IP address matches 1st routing table entry and will be routed to router 1

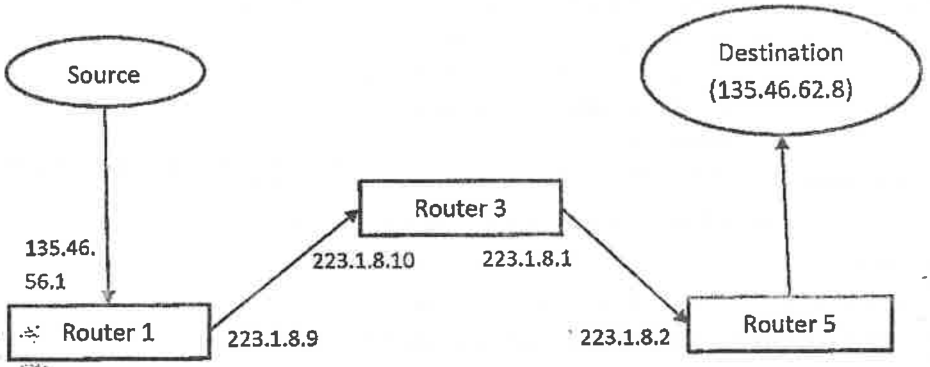
(iii) The first 24 bits (193.53.40) matches the 3rd entry of the routing table. As the 3rd entry indicates the first 23 bits need to be matched, the condition is satisfied and the packet will be routed to router 4.

(iv) The first 16 bits of the IP address (135.46) matches the 1st and 2nd entry of the routing table. Bits 17-22 of the IP address are: 001111. Bits 17-22 of 1st routing table entry are: 001110, bits 17-22 of the 2nd routing table entry are: 001111, therefore the IP address matches 1st routing table entry and will be routed to router 5

(c)

Source IP address 135.46.57.8 belongs to Net A, corresponding to the 1st routing table entry. Destination IP address 135.46.62.8 belongs to Net D, corresponding to the 2nd routing table entry.

Visualize the route taken (not required). all IP addresses pointed by the arrow will be shown on the traceroute printout



Printout:

Tracing route to 135.46.62.8

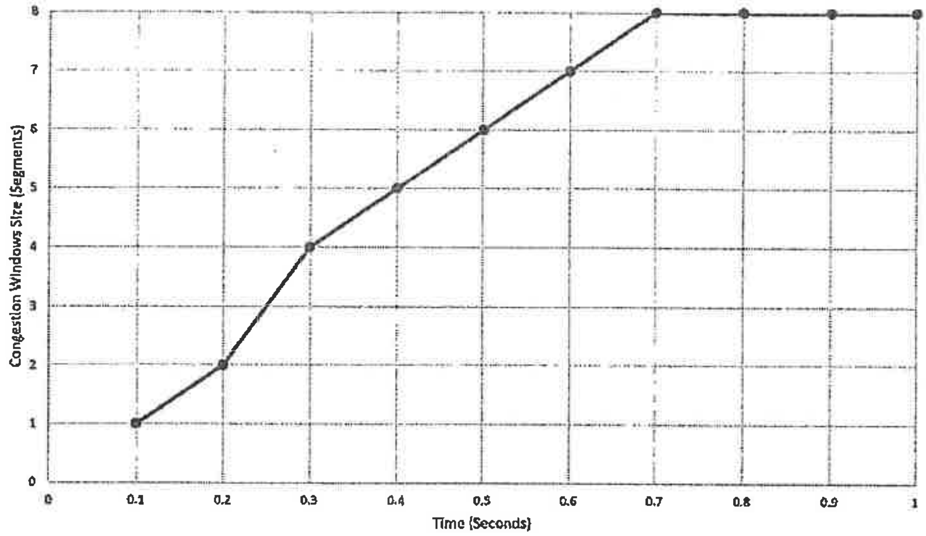
1. 135.46.56.1
2. 223.1.8.10
3. 223.1.8.2
4. 135.46.62.8

Trace complete

4. (a)

Assume that the initial slow start threshold is 4

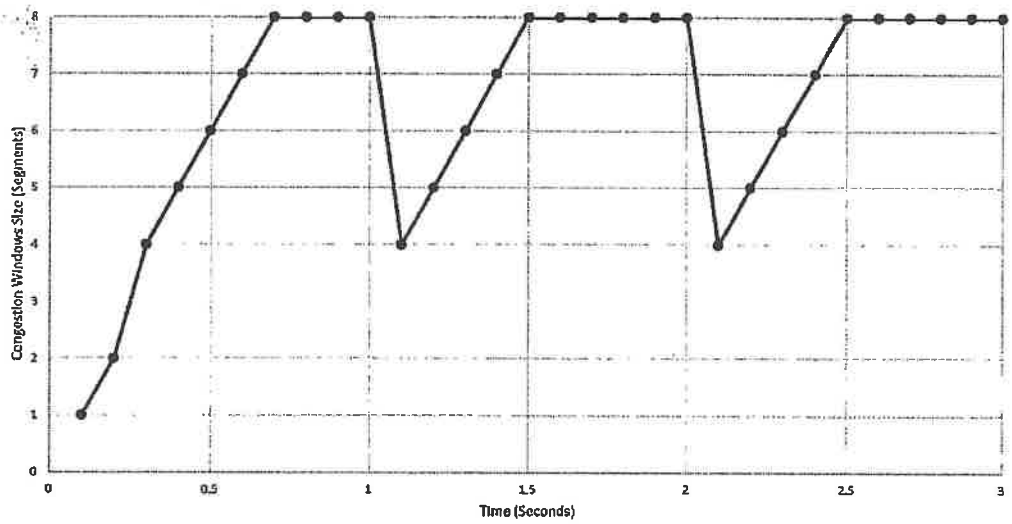
Maximum throughput per round trip is 8 segments and 4KB in total. It takes 4KB/1Gbps = $3.2*10^{-5}$ second <<< RTT = 0.1 second, therefore the transmission delay can be ignored



(b)

Assume TCP Reno is used and the initial slow start threshold is 4. As proved in (a) the transmission delay on this link can be ignored.

Visualize the variation congestion window size for 3 seconds (not required). TCP Reno defines that when triple duplicate ACK is encountered, the congestion window size will reduce to half and start TCP congestion avoidance.



As the maximum throughput of TCP should be the average speed of a large duration of time, therefore we can ignore the slow start section (first second) as it doesn't repeat itself.

For every second, 10 round trips are made and the number of segments sent during each round trip are 4, 5, 6, 7, 8, 8, 8, 8, 8, 8. In total, 70 segments are sent each second. So the maximum throughput of this TCP connection is 70 * 512 = 35840 Bytes per second.

(c)

Note: the file size is corrected to 10GBytes during exam.
TCP sequence number is 32 bit and an increment of 1 means 1 byte is sent out.
Therefore 2^{32} Bytes (= 4 GB) of data need to be sent before the sequence number wraps around.
For part (a), the maximum throughput of TCP (ignore slow start) is 8*512Bytes/0.1second = 40960 Bytes per second.
Therefore, the duration is 2^{32} Bytes/ 40960 Bytes per second = 104857.6 seconds ≈ 29.13 hours.
So it takes 29.13 hours for the sequence number to wrap around.

For reporting of errors and errata, please visit pypdiscuss.appspot.com
Thank you and all the best for your exams! ☺