Networks Assignment 3

Faran Ahmad	Kartikeya Gupta	Ronak Khandelwal
2013CS10220	2013CS10231	2013CS50295

Answer 1 :-

Part i) This is a case of fast retransmit. Three consecutive acknowledgements, expecting the byte number 300 is received by the sender at times 13s/r, 17s/r and 19s/r. As the sender receives three duplicate acknowledgements, it assumes the packet is lost, which triggers the retransmission of the lost packet 300-399.

Part ii) Round Trip Time = 8s/r.

If acknowledgement is received by the sender at 21s/r, then packet corresponding to that will be sent at (21-8)s/r = 13s/r.

Thus, acknowledgement received at 21s/r was generated at the receipt of packet 600-699.

Part iii) Acknowledgement at time 21s/r received by the sender, was sent by the receiver at time 17s/r. At that time the receiver did not receive the packet 300-399, therefore the acknowledgement was sent by the receiver expecting the byte 300 (which is the lost packet).

Part iv) The packet 300-399 was received by the receiver at time 23s/r. Thus, the acknowledgements sent by the receiver at time 17s/r, 19s/r and 21s/r were still expecting byte 300. Correspondingly, these acknowledgements were received by the sender at time 21s/r, 23s/r and 25s/r. The receipt of three duplicate acknowledgements for packet number 300 lead to retransmission of the packet by the sender (according to Fast Retransmission).

Part v) At time 21s/r the receiver had received the packets till byte 899(except the packet 300-399). So, the receiver sends a cumulative acknowledgement expecting the byte 300, which is received by the sender at time 25s/r. However, at time 23s/r the receiver receives the packet 300-399. Therefore at that time, the receiver has received the all packets till 899 and generates acknowledgement expecting the byte 900. This acknowledgement is received by the sender at 27s/r leading to a sudden jump (from 300 at 25s/r to 900 at 27s/r) at that point.

Answer 2:-

Part i) At time 17s/r, 4 acknowledgements have been received by the sender. Therefore, sender increments its window size by 1 (0.25*4). So window size at time 17s/r is 5(4 + 0.25*4).

Part ii)

a) At time 19s/r, the sender has received 3 duplicate acknowledgments for byte 300, there-by reducing the window size to half of the present size. So, the new window size at time 19s/r is 2.5 (5/2).
b) At time 19s/r, 5 acknowledgements have been received by the sender, 3 of which were duplicate acks. Also, the sender has sent 8 packets till the time 19s/r. Therefore, one packet is lost (due to three duplicate acks received) and 2 packets are still in the window because their corresponding acknowledgements has not been received by the sender. Thus, number of outstanding packet is 2. Therefore, no more packets can be sent by the sender because the window size s 2.5.
c) At time 21s/r, one more acknowledgement has been received by the sender.

   Current window size = 2.5

   Increment = 1/int(current window size) = ½ = 0.5.

   Window size at time 21s/r = 2.5 + 0.5 = 3.

d) At time 21s/r, 8 packets have been sent by the sender and acknowledgement for 6 packets have been received with one packet assumed to be lost (because of three duplicate acks). As the sender has not received acknowledgement for one packet that it has sent (packet 700-799), number of outstanding packets is 1.

Pat iii) At time 23s/r, number of outstanding packets is 2 (one sent by the sender at time 17s/r and the other at 21s/r). Also the window size at 23s/r is 3. So, sender can send one more packet at time 23s/r though it didn't receive any acknowledgement at that time.

Part iv) At time 23s/r window size is 3. From the time 23s/r to 31s/r, 3 acknowledgements were received by the sender, none of which are duplicate acks. Therefore, window size at 31s/r is 4 (3 + 1/3 + 1/3 + 1/3).

Answer 3:-

Part i) size of each packet = s.

Round Trip Time = (s/100r + s/(r/2) + s/r) + 4s/r = 7.01s/r.

{Transmission time} + {Propogation time}

Part ii) No packet is dropped till byte 300-399. But the sender receives the acknowledgement for packet 400 consecutively. This implies packet 400-499 is dropped. Therefore, the buffer size is equal to 400.

Part iii)

Time (in s/r)	Window Size Calculation	Window Size
8	8+1/8	8.125
10	8.125 + 1/(int)8.125	8.250
12	8.25 + 1/(int)8.250	8.375
14	8.375 + 1/(int)8.375	8.5
15	8.5 + 1/(int)8.5	8.625
17	8.625/2	4.3125
19	4.3125 + 1/(int)4.3125	4.5625
21	4.5625 + 1/(int)4.5625	4.8125
22	4.8125 + 1/(int)4.8125	5.0625

a) At time 17s/r number of acknowledgements received is 6, number of packets assumed to be lost is 1 (because of three duplicate acks) and total number of packets sent by the sender is 13. Therefore, total number of outstanding packets is 6 (13-6-1). As number of outstanding packets is more than window size no more packets can be sent at 17s/r. Therefore, no retransmission occurs at that time. Till time 22s/r number of outstanding packets is more than window size.
b) At time 22s/r, number of acknowledgements received is 9, number of packets lost is 1 and total number of packets sent is 13. Therefore, number of outstanding packets is 3 (13-9-1). But window size is 5.0625, therefore two more packets can be sent at time 22s/r.

Part iv)

Timeout occurs at 26s/r for packet 500-599. The packet was sent at t = 0, but its acknowledgement was not received till time 26s/r and it was in the window because 3 dup-acks were not received for it. As, at time 26s/r, 25 units of time had passed, the connection was timed out.

Answer 4

i) The round trip time for the packets are as follows:

Packet Number	RTT (in s/r)
1^{st} packet	8
5^{th} packet	16
8^{th} packet	22

ii) The window size is increased to 9 at t=22. This is when the 8^{th} packet gets acknowledged.

iii) The problem with this is:

a) As the window size keeps increasing the chances of a timeout happening will increase and this will hence reduce the window size drastically.

b) As the number of packets in the system keep increasing, it will starve out the packets of other connections at the routers, thus violating TCP friendliness.

iv) A method to trigger a size reduction is:

a) To use the ECN bits. Whenever a router is getting clogged, it should turn the ECN bit on which will reduce the window size of the connection before the actual loss occurs.

b) Another method to trigger this is to limit the window size based on the RTT. If the RTT exceeds a particular threshold, then the window size should be reduced.

Answer 5

Number of packets sent = C + (C+1) + (C+2) + ….. + (2C) = N

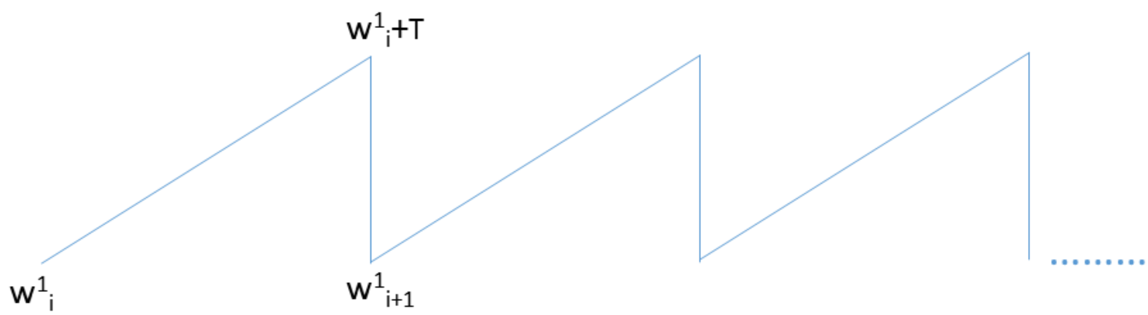
- ⇨ $3C^2 + C - 2*N = 0$
- ⇨ C = (sqrt(24N + 1) -1)/6
- ⇨ Now as L = 1/N
- ⇨ C = (sqrt(24+L) –sqrt(L))/(6*sqrt(L))

Throughput = Number of packets * MSS /(RTT*C)

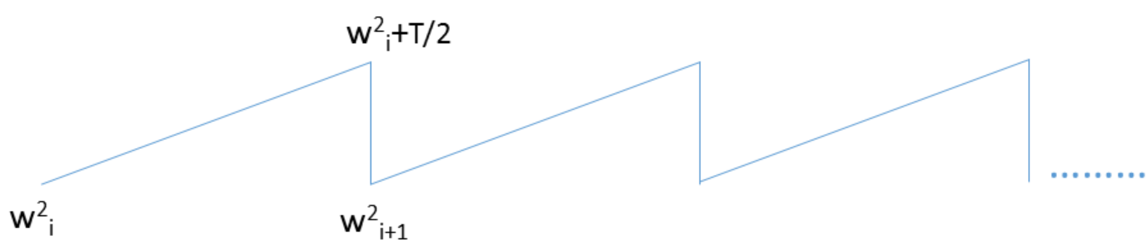
- ⇨ $(3C^2+C)$*MSS/(2*C*RTT)
- ⇨ (3C+1)*MSS/2*RTT
- ⇨ Substituting from above,
- ⇨ (sqrt(24+L)+sqrt(L))*MSS/(4*sqrt(L)*RTT)
- ⇨ Sqrt(24)*MSS/4*sqrt(L)*RTT (As L is very small)
- ⇨ 1.22MSS/RTT*sqrt(L)

Answer 6

Let’s say, RTT of first TCP flow is R, and RTT of other TCP flow is 2R.



First TCP Connection



Second TCP Connection

Since, the buffer drops the packet of both TCP flows at the same time, therefore the congestion window for both the connection become half at the same time.

From the graph,

$w^1_{i+1} = (w^1_i + T)/2$ ---- for 1st TCP connection ---- 1

$w^2_{i+1} = (w^2_i + T/2)/2$ ---- for 2nd TCP connection ---- 2

Let’s say the max. Congestion window is C_{max}

$w^1_i + T + w^2_i + T/2 = C_{max}$

$2*w^1_{i+1} + 2*w^2_{i+1} = C_{max}$

$w^1_{i+1} + w^2_{i+1} = C_{max}/2$ ---- 3

At steady state, for both the connection,

$w_{i+1} = w_i$

Therefore,

$w^1_i + w^2_i = C_{max}/2$ ----- 4

Putting 3 and 4 in eq^n 1 and 2 and adding them,

$C_{max}/2 = C_{max}/4 + (T+T/2)/2$

$T = C_{max}/3$

Therefore, we get

$w^1_{i+1} = (w^1_i + C_{max}/3)/2 = w^1_{i-1}/4 + C_{max}*(1 + ½)$

This goes on and becomes infinite G.P with a= ½ and ratio = ½

We get, $w^1_i = C_{max}/3$ and $w^2_i = C_{max}/6$

Thus, the first connection (with RTT = R) settles at twice the bandwidth as compared to the TCP connection with RTT = 2R.