1. Calculate the sender utilization of a link that connects two devices at 10 Mbps, in which the length of the link is 10 km and the frame size is 1Kilobits. Assume Stop-and-wait protocol for the reliable data transfer protocol (i.e. rdt 3.0). You may ignore the transmission time for the ACK frame in Stop-and-wait. Assume that the speed of light in the medium is 2x108 m/sec.
2. Given,

Distance, d = 10KM = 10 \* 103 mts

Speed of Light, v = 2 \* 108 mts/sec

Frame Size, L = 1Kbits = 1 \* 103 bits

Link Rate, R = 10Mbps = 10 \* 106 bps = 107 bps

Round Trip Time, RTT = 2 \* d / v = 10 ^ (-4) sec

L / R = 10^3 /10^7 = 10 ^ (-4) sec

Usend = (L / R) / (RTT + (L / R))

= (10^ (-4)) / (2 \* 10^ (-4))

= 0.5

Utilization Percentage = 50%

1. In question (1), what should the window size be to achieve maximum sender utilization?
2. For maximum sender utilization the window size should be greater than or equal to [2 \*(RTT/(L/R))] + 1

Hence window size , W = [2 \* (1)] + 1 = 3

1. Given,

SampleRTT values as 106ms, 120ms, 140ms, 90ms, and 115ms

Initial EstimatedRTT = 100ms

α = 0.125

Initial DevRTT = 5ms

β = 0.25

We know that,

EstimatedRTT = (1-α) \* EstimatedRTT + α \* SampleRTT

DevRTT = (1-β) \* DevRTT + β \* | SampleRTT – EstimatedRTT|

TimeoutInterval = EstimatedRTT + 4 \* DevRTT

For 106ms:

EstimatedRTT = 0.875 \* 100 + 0.125 \*106 = 100.75ms

DevRTT = 0.75 \* 5 + 0.25 \* | 106 – 100.75 | = 5.06ms

TimeoutInterval = 100.75 + 4 \* 5.06 = 120.99ms

For 120ms:

EstimatedRTT = 0.875 \* 100.75 + 0.125 \*120 = 103.15 ms

DevRTT = 0.75 \* 5.06 + 0.25 \* | 120 – 103.15 | = 8ms

TimeoutInterval = 103.15 + 4 \* 8 = 135.15ms

For 140ms:

EstimatedRTT = 0.875 \* 103.15 + 0.125 \*140 = 107.76ms

DevRTT = 0.75 \* 8 + 0.25 \* | 140 – 107.76 | = 14.06ms

TimeoutInterval = 107.76+ 4 \* 14.06 = 164ms

For 90ms:

EstimatedRTT = 0.875 \* 107.76 + 0.125 \*90 =105.54ms

DevRTT = 0.75 \* 14.06 + 0.25 \* | 90 – 105.54 | =14.43ms

TimeoutInterval = 105.54 + 4 \* 14.43 = 163.26ms

For 115ms:

EstimatedRTT = 0.875 \* 105.54 + 0.125 \*115 = 106.72ms

DevRTT = 0.75 \* 14.43 + 0.25 \* | 115 – 106.72 | = 12.89ms

TimeoutInterval = 106.72 + 4 \* 12.89 = 158.28ms

1. Host A and B are communicating over a TCP connection in which the maximum segment size is 1024 bytes, and the receiver window size is 4KB. If A has transmitted 2048 bytes which have been successfully acknowledged by B,

1. What are the value(s) of the sequence number(s) for the packet(s) that A sends to B, if the application running at A writes 8 KB data?

2. What is the value of the acknowledgement field for the last packet that B sends to A, assuming that B’s receive window is full?

A. Given,

Maximum Segment Size = 1024 bytes

Receiver Window Size = 4KB = 4000 bytes

Application at Host A sends 8KB of data i.e. 8000 bytes

Since the receiver window size is less than the data sent by the sender the sequence numbers will be regenerated from the beginning again once the receiver window is full.

As Maximum segment size is 1024 bytes we divide the data of 8000 bytes into 1024 bytes for each packet

Packet1 = 1024 bytes

Packet2 = 1024 bytes

Packet3 = 1024 bytes

Packet4 = 928 bytes

Packet5 = 1024 bytes

Packet6 = 1024 bytes

Packet7 = 1024 bytes

Packet8 = 928 bytes

As already 2048 bytes of data has been received by the Host B the sequencing

for the first packet will start from 2048.

Sequence number for the first acknowledgement = 2048+1024 = 3072

For the second packet = 3072 + 1024 = 4096

For the third packet = 4096 + 1024 = 5120

For the fourth packet = 5120 + 928 = 6048

Hence the sequence numbers for packet A are 3072, 4096, 5120, 6048

The value of the Acknowledgment field for the last packet that B sends to A is 6048

1. The retransmitted segments contain the same data and the same sequence number, so there will be the same ACK numbers as the previously sent TCP segments. It is possible that after a very short time, we receive the ACK for the retransmitted data segments. But that can be the delayed ACK for the original TCP segment. If we count this time in to compute the RTT estimation, there might be a possibility that the RTT average will be dropped by big percentage. This would make the retransmission faster that in turn makes the network busy.
2. c) After the 16th transmission round, the packet loss is recognized by a triple duplicate ACK. If there is a timeout, the congestion window size would have dropped by 1.

e) The threshold is initially 32, since it is at this window size that slow start stops and congestion avoidance begins.

1. the threshold will be set to half the current value of the congestion window(8), when the loss occurred and congestion window will be set to new threshold +3 MSS. TCP halves the value of cwnd but adds 3 MSS for good measure to account for triple duplicate ACKS received . Thus the new values for congestion window and ssthresh will be 7 and 4 respectively.
2. Given,

a) W \* MSS/RTT = 10Mbps = 10 \* 10^ (6) = 10^ (7) bytes

W \* 1500\*8 / 0.15 = 10^ (7)

W = 1000 / 8

W = 125

c) (windowsize / 2) \* 0.15 = 9.375sec, as the number of RTTs is given by W/2.