Problem 1

Suppose that instead of a multiplicative decrease, TCP decreased the window size by a constant amount. Would the resulting AIAD algorithm converge to an equal share algorithm? Justify your answer using a graphical diagram similar to Slide 100 of the lecture.

Refer to Figure 5. In Figure 5(a), the ratio of the linear decrease on loss between connection 1 and connection 2 is the same - as ratio of the linear increases: unity. In this case, the throughputs never move off of the AB line segment. In Figure 5(b), the ratio of the linear decrease on loss between connection 1 and connection 2 is 2:1. That is, whenever there is a loss, connection 1 decreases its window by twice the amount of connection 2. We see that eventually, after enough losses, and subsequent increases, that connection 1's throughput will go to 0, and the full link bandwidth will be allocated to connection 2.

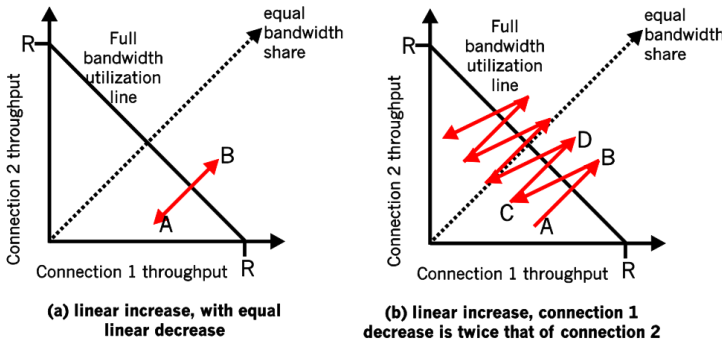


Figure 5: Lack of TCP convergence with linear increase, linear decrease

Problem 2

TCP is a very symmetric protocol, but the client/server model is not. Consider an asymmetric TCP-like protocol in which only the server side is assigned a port number visible to the application layers. Client-side sockets would simply be abstractions that can be connected to server ports. Can you propose header data and connection semantics to support this. What will you use to replace the client port number?

Port number of client side is useful information by which the traffic can be demultiplexed within the network. Demultiplexing traffic is equivalent to distinguish traffic according to its source-destination pair. Here, as the clients are not assigned any ports. there has to be a mechanism in which port number is replaced by some other number. This number which is used on behalf of the port number will be assigned by the client for the connection. Port would not be directly visible anymore. In this case, the header will not change at all as the port number is just replaced by some other number. So, the name of the field has changed only.

Problem 3

On the TCP throughput, in the period of time from when the connections rate varies from W/(2 RTT) to W/RTT, only one packet is lost (at the very end of the period).

(a) Show that the loss rate (fraction of packets lost) is equal to L=lossrate= $1/(3/8W^2 + 3/4W)$

(b) Use the result above to show that if a connection has loss rate L, then its average rate is approximately given by $\simeq 1.22 \times MSS/(RTT \times \sqrt{L})$

a) The loss rate, L, is the ratio of the number of packets lost over the number of packets sent. In a cycle, 1 packet is lost. The number of packets sent in cycle is

$$\begin{aligned} \frac{W}{2} + (\frac{W}{2} + 1) + ... + W &= \sum_{n=0}^{W/2} (\frac{W}{2} + n) \\ &= (\frac{W}{2} + 1)\frac{W}{2} + \sum_{n=0}^{W/2} n \\ &= (\frac{W}{2} + 1)\frac{W}{2} + \frac{W/2 \cdot (W/2 + 1)}{2} \\ &= \frac{W^2}{4} + \frac{W}{2} + \frac{W^2}{8} + \frac{W}{4} \\ &= \frac{3}{8}W^2 + \frac{3}{4}W \end{aligned} \tag{1}$$

Thus the loss rate is

$$L = \frac{1}{\frac{3}{8}W^2 + \frac{3}{4}W} \tag{2}$$

b) For W large, $\frac{3}{8}W^2 >> \frac{3}{4}W$. Thus $L \approx \frac{8}{3W^2}$ or $W \approx \sqrt{\frac{8}{3L}}$. From the text, we therefore have

$$\begin{aligned} averagethroughput &= \frac{3}{4}\sqrt{\frac{8}{3L}}\frac{MSS}{RTT} \\ &= \frac{1.22MSS}{RTT\sqrt{L}} \end{aligned} \tag{3}$$

Problem 4

You are designing a reliable, sliding window, byte-stream protocol similar to TCP. It will be used for communication with a geosynchronous satellite network, for which the bandwidth is 1 Gbps and the RTT is 300 ms. Assume the maximum segment lifetime is 30 seconds.

(a) How many bits wide should you make the ReceiveWindow and SequenceNum fields? (ReceiveWindow is also called "Advertised Window" in some other textbooks.)

(b) If ReceiveWindow is 16 bits, what upper bound would that impose on the effective bandwidth?

(a) To fully utilize the network, ReceiveWindow needs to be larger than (Delay × Bandwidth). (ReceiveWindow) ≥ (Delay) × (Bandwidth) = 300 ms × 1 Gbps = 300 Mbit = 37.5 MB

$2^{25} = 33,554,432$

$2^{26} = 67,108,864$

Therefore, 26 bits are needed for ReceiveWindow.

SequenceNum needs be large enough that the sequence number does not wrap around before any delayed segments have left the network, which is presumed to occur within the maximum segment lifetime.

SequenceNum ≥ (Maximum Segment Lifetime) × (Bandwidth) = 30 s × 1 Gbps = 30 Gbit = 3.75 GB

$2^{31} = 2,147,483,648$

$2^{32} = 4,294,967,296$

Therefore, 32 bits are needed for SequenceNum.

(b) If ReceiveWindow is 16 bits, it is smaller than (Delay × Bandwidth) product. Therefore, ReceiveWindow is the limiting factor of the effective bandwidth. During a single RTT, only ReceiveWindow amount of data can be transferred.

Effective Bandwidth = ReceiveWindow / RTT = 2^{16} B / 300 ms = 218.3 KB/s

Problem 5

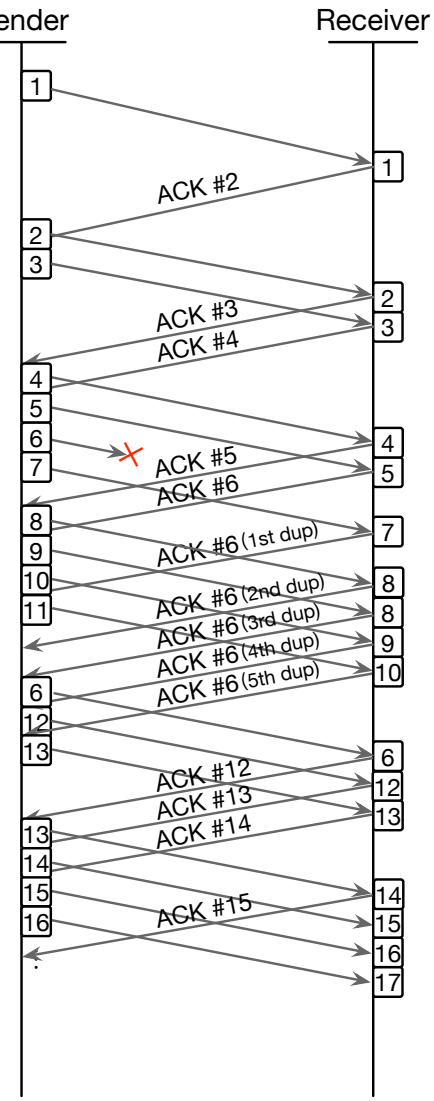
Consider the evolution of a TCP connection with the following characteristics. Assume that all the following algorithms are implemented in TCP congestion control: slow start, congestions avoidance, fast retransmit and fast recovery, and retransmission upon timeout. If ssthresh equals to cwnd, use the slow start algorithm in your calculation.

- The TCP receiver acknowledges every segment, and the sender always has data segments available for transmission.
- The RTT is 100 ms for all transmissions, consists of the network latency of 60 ms in sending a segment (header and payload) from the sender to the receiver and 40 ms in sending an acknowledgment (header only) from the receiver to the sender. Ignore packet-processing delays at the sender and the receiver.
- Initially ssthresh at the sender is set to 5. Assume cwnd and ssthresh are measured in segments, and the transmission time for each segment is negligible.
- Retransmission timeout (RTO) is initially set to 500ms at the sender and is unchanged during the connection lifetime.
- The connection starts to transmit data at time t = 0, and the initial sequence number starts from 1. TCP segment with sequence number 6 is lost once (i.e., it sees segment loss during its first transmission). No other segments are lost during transmissions.

What are the values for cwnd and ssthreshold when the sender receives the TCP ACK with number 15? Show your intermediate steps or your diagram in your solution.

Problem 5 continued on next page...

Algorithm	ssthresh	cwnd	window
slow start	5	1	1
slow start	5	1 + 1 = 2	3 2
slow start	5	2 + 1 = 3	5 4 3
slow start	5	3 + 1 = 4	7 6 5 4
slow start	5	4 + 1 = 5	9 8 7 6 5
slow start	5	5 + 1 = 6	11 10 9 8 7 6
no action	5	6	11 10 9 8 7 6
no action	5	6	11 10 9 8 7 6
FR & FR	$\lfloor 6 / 2 \rfloor = 3$	3 + 3 = 6	11 10 9 8 7 6
FR & FR	3	6 + 1 = 7	12 11 10 9 8 7 6
FR & FR	3	7 + 1 = 8	13 12 11 10 9 8 7 6
FR & FR → SS	3	3 → 3 + 1 = 4	15 14 13 12
congestion avoidance	3	$\lfloor 4 + 1/4 \rfloor = 4$	16 15 14 13
congestion avoidance	3	$\lfloor 4.25 + 1/4 \rfloor = 4$	17 16 15 14
congestion avoidance	3	$\lfloor 4.5 + 1/4 \rfloor = 4$	18 17 16 15



When ACK #15 is received, ssthresh = 3 and cwnd = 4