Problem 1

We have discussed Karn's algorithm in the lecture. Why do you think TCP avoids measuring the SampleRTT for retransmitted segments? Please state what are penalties if the ACK is considered as that for the originally transmitted segment, or for the retransmitted segment.

Write your solution to Problem 1 in this box

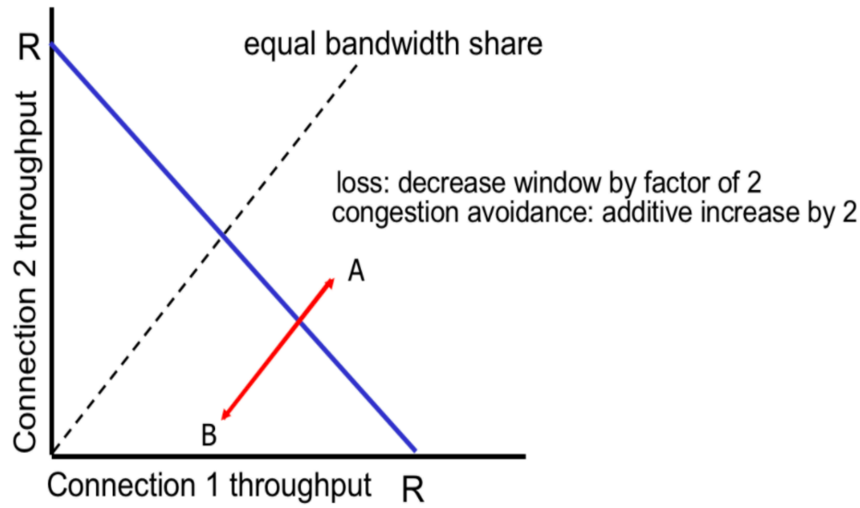
1. For retransmit package, usually it happens due to packet loss. TCP avoid measuring SampleRTT for retransmitted segments because in retransmitting the same packet, it will have a same ACK with the same sequence number. In addition, there is a chance of delay ACK upon receiving the retransmitted packet. If TCP count this sampleRTT, it will have a miscalculation on predicting the RTT estimation. Not only that, retransmit ACK on a retransmit packet will improve the probability of the network to get congested.

Problem 2

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.

Write your solution to Problem 2 in this box

2. If TCP decreased the window size by a constant amount, then the final result of the AIAD algorithm will never converge to an equal bandwidth share. It will just go back and forth between A and B (in diagram provided below). Thus, it will never approach equal bandwidth share.



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 $\simeq 1.22 \times MSS/(RTT \times \sqrt{L})$

Write your solution to Problem 3 in this box

3a. $\frac{W}{2} + \left(\frac{W}{2}+1\right) + \dots + W = \sum_{n=0}^{W/2} \left(\frac{W}{2} + n\right)$

$$= \left(\frac{W}{2}+1\right) \cdot \frac{W}{2} + \sum_{n=0}^{W/2} n$$

$$= \left(\frac{W^2}{4} + \frac{W}{2}\right) + \frac{W/2 \cdot (W/2+1)}{2}$$

$$= \frac{W^2}{4} + \frac{W}{2} + \frac{W^2}{8} + \frac{W}{4}$$

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

$\therefore$ Loss rate is $\dfrac{1}{\frac{3}{8}W^2 + \frac{3}{4}W}$

b. If W is large, $\frac{3}{8}W^2 > \frac{3}{4}W$, thus,

$L \approx \frac{8}{3}W^2 \rightarrow W \approx \sqrt{\frac{8}{3L}}$

$\therefore$ avg throughput $= \frac{3}{4}\sqrt{\frac{8}{3L}} \cdot \frac{MSS}{RTT}$

$$= \frac{1.22\ MSS}{RTT\sqrt{L}}$$

Problem 4

The value of the threshold ssthresh is set as ssthresh = cwnd/2 in TCP congestion control, which is referred to as being set to half the window size when a loss event occurred. Must the rate at which the sender is sending when the loss event occurred be approximately equal to cwnd segments per RTT? Explain your answer. If your answer is no, can you suggest a different manner in which ssthresh should be set?

Write your solution to Problem 4 in this box

4. Yes, ssthresh must be = cwnd/RT because cwnd uses to limit the window of the sender size, which is going to be divided by 2 when there is a congestion or packet loss to avoid the congestion getting worse. In addition, ssthresh also used to store the second state value, and it's reset when segment loss occurred or there is a congestion. If it is not equal to the number of RTT, it's going to be hard when to determine when is the congestion or packet loss occurred. Thus, it needs to be cwnd/RTT.

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.

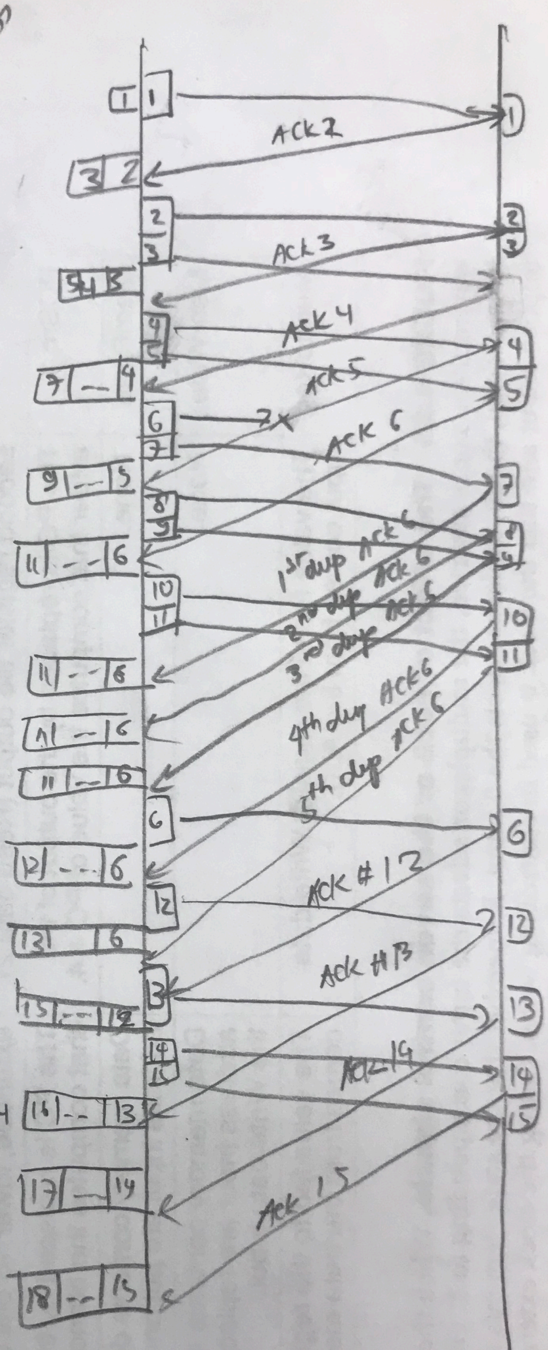
- The TCP receiver acknowledges every segment, and the sender always has data segments available for transmission.
- The network latency in sending a segment (header and payload) from the sender to the receiver is 30ms and the network latency in sending an acknowledgment (header only) from the receiver to the sender is 20ms. 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 RTT is 100ms for all transmissions.
- 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. . .

Write your solution to Problem 5 in this box

algorithm	ssthrsh	cwnd
SS	5	1
SS	5	2
SS	5	3
SS	5	4
SS	5	5
SS	5	6
CA	5	6
CA	5	6
FR	3	6
FR	3	7
FR	3	8
CA	3	4
CA	3	$4+\frac{1}{4} \approx 4$
CA	3	$4+\frac{2}{4} \approx 4$
CA	3	$4+\frac{3}{4} \approx 4$



sstresh at ack 15 is 3 and the cwind is 4