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

Retransmitted packages/segments are exactly the same as the original segment, including the sequence number. That is to say the ACKs for both segments are also the same, and the sender cannot tell the difference.

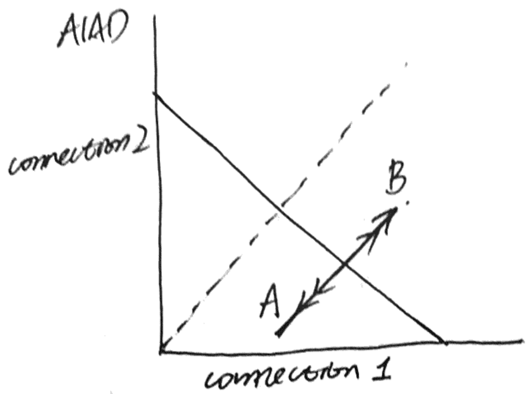
When both segments are delayed through the network and the first ACK (for the original segment) arrives at the sender after the retransmission, the sender would assume this ACK is for the retransmitted segment and thus miscalculate the SAMPLERTT. ~~to~~ It's gonna be mistakenly lower than the correct RTT, and the sender will send faster ~~retransmission~~ which makes the jammed network ~~~~ even worse.

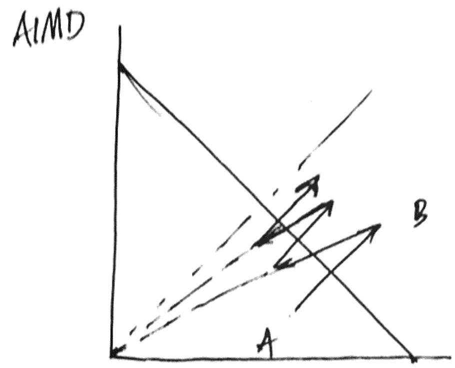
Problem 2

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

AIAD algorithm does not acheive fairness.
It will not converge to an equal share algorithm





Additive increase gives slope of 1,
So is additive decrease.
Therefore, no matter ~~fo~~ increase ~~or~~ or decrease,
it's always going to be ~~in the AB the direction~~ along AB.
Line AB is parallel with the "fairness line",
Thus AIAD algorithm never converges to equal share.
(Oscillation along AB ~~continues~~ keeps going forever)

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

(a) loss rate is [# of packets lost] / [# of packets sent]

$$[\text{\# of packets sent}] = \left(\frac{W}{2}\right) + \left(\frac{W}{2}+1\right) + \dots + W = \sum_{n=0}^{W/2}\left(\frac{W}{2}+n\right)$$

$$= \sum_{n=0}^{W/2}\frac{W}{2} + \sum_{n=0}^{W/2} n$$

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

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

$$L = \frac{1}{[\text{\# of packets sent}]} = \boxed{\frac{1}{\frac{3}{8}W^2 + \frac{3}{4}W}}$$

(b) if W is large, then $\frac{3}{8}W^2 >> \frac{3}{4}W$

i.e. $L \fallingdotseq \frac{1}{\frac{3}{8}W^2} \Rightarrow W \fallingdotseq \sqrt{\frac{8}{3L}}$

therefore, $[\text{Average throughput}] = \frac{3}{4}\cdot W \cdot \frac{MSS}{RTT}$

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

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

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

Yes. rate = cwnd/RTT when a loss event occurs.

cwnd refers to congestion window size, and it's designed for constraining sender's sending rate.

SSthresh is used to store the second state value. it's reset to cwnd/2 once a loss event is identified

Therefore the sender rate must be approximately equivalent to cwnd/RTT when a loss event occurs.

So ~~the~~ statement is true.

Problem 5

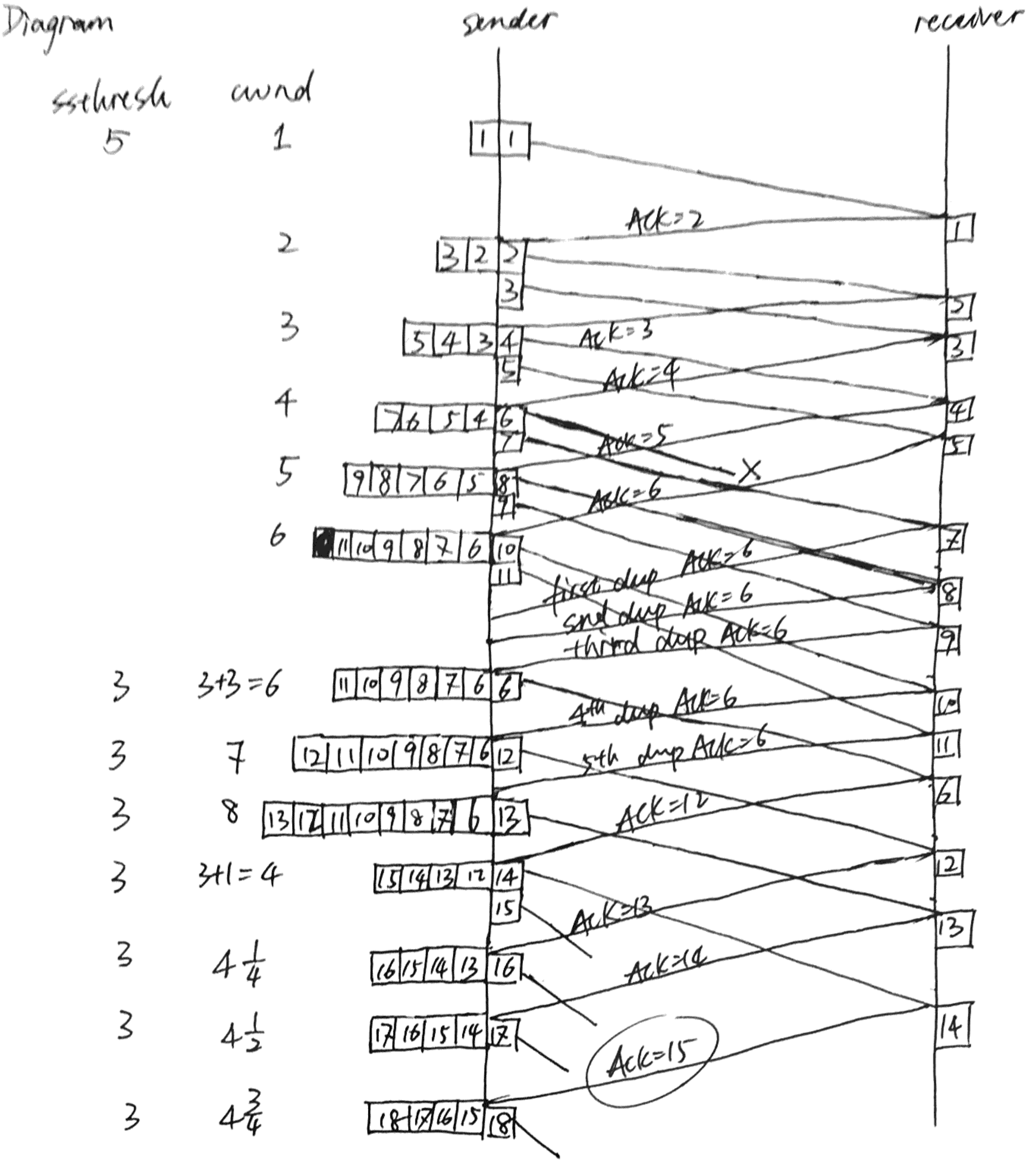
Consider the evolution of a TCP connection with the following characteristics. Assume that all the following algorithms are implemented in TCP congestion control: slow start, congestions avoidance, fast retransmit and fast recovery, and retransmission upon timeout. If ssthresh equals to cwnd, use the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

Write your solution to Problem 5 in this box

Diagram



When the sender receives ACK 15,

cwnd = $4\frac{3}{4}$

ssthresh = 3