

# 15-441: Computer Networks

## Homework 2 Solution

Assigned: September 25, 2002.  
Due: October 7, 2002 in class.

In this homework you will test your understanding of the TCP concepts taught in class including flow control, congestion control, and reliability.

**You must solve the homework individually. Make sure you provide all your answers in the space provided in the end. For full credit, show your work in clearly readable writing. Points will be deducted for unreadable answers. The following notation should be used for all the calculations.**

Parameter	Description	Units
$MSS$	Maximum Segment Size	bytes/packet
$W$	Window Size	packets
$W * MSS$	Window Size	bytes
$RTT$	Round trip time	seconds
$X$ or $B$	Throughput	bits/sec
$\rho$	Utilization	dimensionless percentage
$L$	Loss rate	dimensionless proportion/probability

Table 1: Notation of parameters to be used throughout the assignment

## 1 TCP Basics

### 1. True or False?

- (a) Host A is sending a large file to host B over a TCP connection. Assume host B has no data to send to host A. Host B will not send acknowledgments to host A because host B cannot piggyback the acknowledgments on data.

**Answer:** FALSE. Piggybacking is an optimization that is used when both sides have to send data to each other so that the receiver, instead of sending two packets i.e., an ACK and a data packet, it just sends one. When the receiver (B) does not have any data to send, it will still send an ACK with the sequence number field containing the next sequence of data it is supposed to send.

- (b) The size of the TCP RcvWindow never changes throughout the duration of the connection.

**Answer:** FALSE. RcvWindow is used to give the sender an idea of how much free buffer space is available at the receiver. Thus as the amount of unacknowledged TCP data varies the RcvWindow also changes.

- (c) Suppose host A is sending a large file to host B over a TCP connection. The number of unacknowledged bytes that A sends cannot exceed the size of the advertised receiver buffer.

**Answer:** TRUE. TCP is not permitted to overflow the allocated receiver buffer. Hence when the sender can not send any more data RcvWindow would be 0 and hence all the buffer would have unacknowledged data.

- (d) Suppose that the last `SampleRTT` in a TCP connection is equal to 1sec. Then the current value of `TimeoutInterval` for the connection will necessarily be  $\geq 1$  sec.

**Answer:** FALSE. Depending on the value of  $\alpha$  or `EstimatedRTT` it may or may not be greater than 1.

2. Suppose host A is sending a large file to host B over a TCP connection. The two end hosts are 10msec apart (20msec RTT) connected by a 1Gbps link. Assume that they are using a packet size of 1000 bytes to transmit the file. Also assume for simplicity that ACK packets are extremely small and can be ignored.

- (a) At least how big would the window size (in packets) have to be for the channel utilization to be greater than 80%.

**Answer:** Bandwidth-delay product of A to B =  $10^9 \times 20 \times 10^{-3} = 2 \times 10^7$  bits.

For 80% utilization, it will be  $2 \times 10^7 \times 0.8 = 1.6 \times 10^7$  bits.

Number of packets =  $\frac{1.6 \times 10^7}{8 \times 1000} = 2000$  packets.

- (b) Assuming infinite initial threshold, no losses and competing traffic, approximately how long (in seconds) would it take for the normal slow start mechanism to achieve 80% utilization?

**Answer:** Number of RTTs =  $\lceil \log_2(2000) + 1 \rceil = 12$ .

Time =  $12 \times 20 \times 10^{-3} = 0.24$  sec.

3. TCP waits until it has received three duplicate ACKs before performing a fast retransmit. Why do you think the TCP designers chose not to perform a fast retransmit after the first duplicate ACK for a segment is received?

**Answer:** Packets can arrive out of order from the IP layer. So whenever an out of order packet would be received it would generate a duplicate ACK and if we perform retransmission after the first duplicate ACK it would lead the sender to introduce too many redundant packets in the network.

## 2 TCP Security

Consider a misbehaving TCP receiver. The receiver modifies its TCP such that upon receiving a data segment containing  $N$  bytes, the receiver divides the resulting acknowledgment into  $M$ , where  $M \leq N$ , separate acknowledgments each covering one of  $M$  distinct pieces of the received data segment. For e.g. if it receives data acknowledging bytes 1 to 1000, then the receiver, for  $M = 2$ , will send 2 ACKs for 501 and 1001.

Consider a normal TCP sender sending data to this misbehaving TCP receiver. The sender sends a 1500 byte packet with sequence number 1. The receiver sends back  $M = 3$  ACKs.

1. What packets will the sender send next in response to the 3 ACKs?

**Answer:** Response will be packets with byte sequence numbers 1501, 3001, 4501 and 6001.

2. Assuming no losses and negligible packet transmission time, derive an expression for the sender window size during the slow start phase, in terms of  $n$  (number of RTTs) and  $M$ .

**Answer:** For each outgoing packet, the receiver sends back  $M$  ACKs. The sender will increment the `cwnd` by  $M$ . So if initially the window was 1, after 1 RTT the window size will be  $M + 1$ . In the second RTT, the sender sends  $M + 1$  packets (the window size) and receives  $M \times (M + 1)$  ACKs. So now the window will increase by  $M \times (M + 1)$  and become  $M \times (M + 1) + M + 1 = (M + 1)^2$ . After  $n$  RTTs, the window size will be  $(M + 1)^n$ .

3. Can you think of a simple enhancement to the sender which can prevent the receiver from launching this attack?

**Answer:** One solution is that the sender should advance the congestion window at byte granularity rather than packet granularity. Hence `cwnd` will be incremented not by a full MSS but only proportional to the amount of data acknowledged. Another solution is to increment `cwnd` by one MSS when a valid ACK arrives covering the entire data segment sent.

### 3 RED/AQM

Consider a bursty greedy source. Would RED drop more packets for this source or Droptail? Explain briefly your answer.

**Answer:** Bursty traffic suffers more losses with drop tail than RED gateways. Consider the case when the queue gets almost full. In droptail gateways an arrival of a burst would result in several losses as most of the burst will be lost. In contrast in RED gateways, it drops packets early so as not to allow the queue to grow fully. So when a burst arrives, it will only drop some of the packets of the burst. The number of packets dropped will be roughly proportional to their share of the bandwidth. Hence the bursty flow should lose fewer packets in RED.

### 4 Fast Recovery and TCP Modelling

**Note: Remember to use Table 1 for all symbols and units. Points will be lost for incorrect units.**

1. Harry Bovik says that when during steady state, achieved TCP throughput should be 75% of the bottleneck bandwidth. His reasoning is that when the throughput reaches 100% of the bottleneck, there is a packet drop causing the window to be halved to 50% and then it increases linearly back to 100% repeating the same cycle. Is he right or wrong? Explain briefly your answer.

**Answer:** He is wrong. The achieved throughput should be close to 100% because of router buffers. As the TCP connection starts sending at rates higher than the bottleneck bandwidth the buffers start to fill up until there is a packet loss. Now the TCP will reduce its rate to 50% of that high rate and so on the average TCP will be sending at much higher rate than 75% of the bottleneck bandwidth.

2. Let  $w$  represent the congestion window size in TCP during steady state. The idealized congestion window size varies from  $\frac{W}{2}$  to  $W$  during steady state and then a loss occurs. Suppose an additional constraint is added such that the maximum receiver window size for a connection is set to  $\frac{3}{4}W$ . Assuming a loss occurs when the congestion window has a size of  $W$ ,

- (a) Find the loss rate  $L$  for this TCP connection in terms of  $W$ . **Hint : Calculate the number of packets sent per loss cycle**

**Answer: Solution 1 (correct):**  $\frac{3W}{8}$  to  $\frac{3W}{4}$

Approximate Number of Packets sent = Area under the graph = rectangular region + trapezium region

$$= \frac{3W}{8} \times \frac{5W}{8} + \frac{1}{2} \left( \frac{5W}{8} + \frac{W}{4} \right) \times \frac{3W}{8} = \frac{51}{128} W^2$$

$$\text{Loss rate } L = \frac{\frac{1}{128} W^2}{\frac{51}{128} W^2} = \frac{1}{51} W^2$$

**Solution 2 (accepted with full points):**  $\frac{W}{2}$  to  $\frac{3W}{4}$

Approximate Number of Packets sent = Area under the graph = rectangular region + trapezium region

$$= \frac{W}{2} \times \frac{W}{2} + \frac{1}{2} \left( \frac{W}{2} + \frac{W}{4} \right) \times \frac{W}{4} = \frac{11}{32} W^2$$

$$\text{Loss rate } L = \frac{\frac{1}{32} W^2}{\frac{11}{32} W^2} = \frac{1}{11} W^2$$

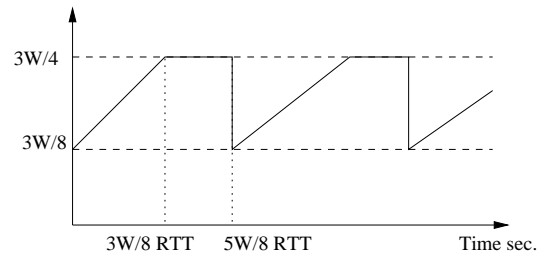
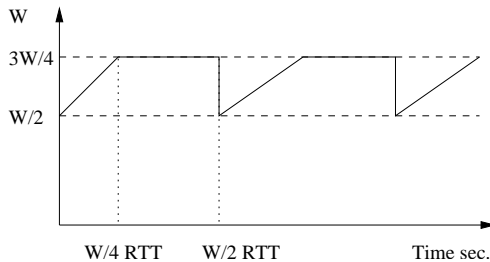


Figure 1: Answer 4.3 : TCP Window with flow control

- (b) Derive an expression for the achieved throughput  $B$  in terms of  $RTT$ ,  $L$  and  $MSS$ .

**Answer: Solution 1 (correct):**  $\frac{3W}{8}$  to  $\frac{3W}{4}$

$$\text{From previous answer, } W = \sqrt{\frac{128}{51L}}$$

$$B = \frac{\text{bitssent}}{\text{time}} = \frac{\frac{51}{128} W^2 \times MSS \times 8}{\frac{5W}{8} \times RTT} = \frac{51W \times MSS}{10 \times RTT}$$

Replacing  $W$  with  $\sqrt{\frac{128}{51L}}$  and simplifying we get

$$B = \frac{4\sqrt{102} \times MSS}{5\sqrt{L} \times RTT} \text{ bps}$$

**Solution 2 (accepted with full points):**  $\frac{W}{2}$  to  $\frac{3W}{4}$

From previous answer,  $W = \sqrt{\frac{32}{11L}}$ .

$$B = \frac{\text{bitssent}}{\text{time}} = \frac{\frac{11}{32} W^2 \times MSS \times 8}{\frac{W}{2} \times RTT} = \frac{11W \times MSS}{2 \times RTT}$$

Replacing  $W$  with  $\sqrt{\frac{32}{11L}}$  and simplifying we get

$$B = \frac{2\sqrt{22} \times MSS}{\sqrt{L} \times RTT} \text{ bps}$$

## 5 Congestion Control

Harry Bovik thinks that if the sources in the Internet use additive increase additive decrease (AIAD) the system should still converge to fairness and efficiency.

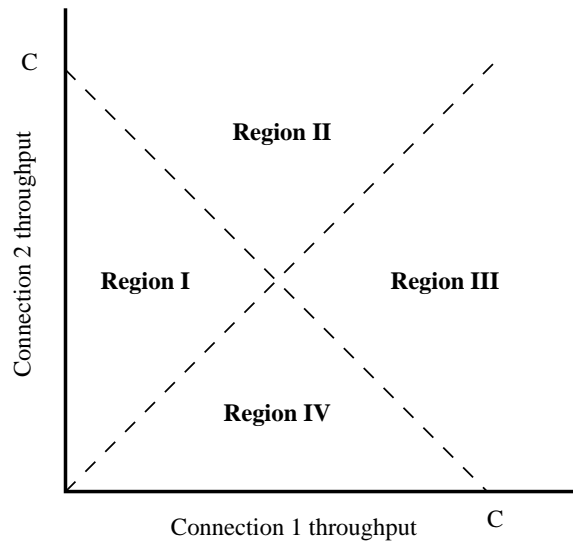


Figure 2: TCP Phase-plot

Harry decides to use phase-plots (see pg. 269 of the book) to check his intuition for the two-user case. Figure 2 shows a simple phase plot. In the graph, the state of the system is represented by a point  $(x_1, x_2)$ , where  $x_1$  is stream 1's current rate and  $x_2$  is stream 2's current rate. The total capacity of the link is  $C$  bits/s. Assume that after the packet loss the window sizes have been reduced according to AIAD and now the system state is in either one of the labelled regions.

1. On the figure, for each of the labelled regions, I, II, III, and IV, draw vectors indicating the direction in which the system will move next i.e., after the AIAD response to the packet loss.

**Answer:** See Figure 3.

2. In each of the following regions, does the response after the packet loss increase, reduce or keep fairness the same?

- Region I: **Answer:** Increases
- Region II: **Answer:** Decreases

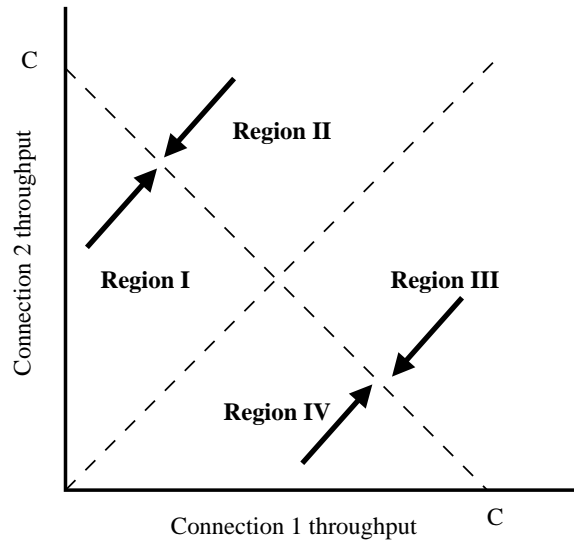


Figure 3: Answer 5.1 : TCP Phase-plot

- Region III: **Answer:** *Decreases*
- Region IV: **Answer:** *Increases*
- On the 45 degree line through the origin: **Answer:** *Unchanged*

3. Does Harry's scheme converge to a fair allocation? Explain your answer briefly.

**Answer:** *No it does not. Both the connections will decrease their share equally on a packet loss (additive decrease) rather than in proportion to their share of C (multiplicative decrease). Similarly both the connections will attempt to grab bandwidth equally when it is available. So the movement will remain on the 45 degree line parallel to the fairness line based on their initial starting point.*

4. Does Harry's scheme converge to an efficient allocation? Explain your answer briefly.

**Answer:** *Yes it does. They adaptively increase their share when they are below the efficiency line (Regions I and IV) to move towards the efficiency line. Similarly they decrease their share on losses and hence move towards the efficiency line when they are above the efficiency line (Regions II and III).*

## 6 Multimedia

1. Harry Bovik decides that for his video application to recover from losses over the Internet he should set the video buffer playout time equal to the current RTT to the sender. He runs his multimedia application playing videos from all around the world. He notices that for larger RTT values his video works fine while for shorter RTT values his video misses several frames. What is the mistake that Harry is making?

**Answer:** *The buffer playout time should be based on the delay jitter rather than end-to-end delay (RTT) to ensure smooth playing. In this case what is happening is that for shorter RTTs the jitter is more than the RTT and these delayed packets result in missed frames while for large RTT values, the jitter is smaller than the RTT and so fewer packets are missed.*

2. Why do you think TCP is not used for streaming multimedia (audio, video) files?

**Answer:** *Providing timely delivery is more crucial in multimedia applications than providing reliability. The multimedia applications can tolerate occasional losses without any need to retransmit all the lost packets. In regards to congestion control, multimedia applications usually have their own application level mechanisms to handle congestion because it often requires understanding of the application data (e.g. B, I and P frames in mpeg) and TCP is not a suitable layer for providing such congestion control mechanisms.*

3. Harry Bovik says that since there are negligible losses within the US, the streaming multimedia applications do not need to have a playback buffer in US. What would be your response?

**Answer:** *The playback buffer is required to smooth out the video. Delay variance causes packets to arrive after their playout time and hence are missed. The playback buffer is needed even if there are no losses.*

## 7 Reliability

For all parts of this question, refer to figure 4. The graph shows the packets sent and ACKs received by a TCP Reno sender. The x-axis shows the time in seconds and the y-axis shows the sequence number of the packet (or ACK) being sent (or received). Assume that all packets that are not lost will arrive in order at both the sender and receiver. Also assume that receiver keeps all the out of order received packets.

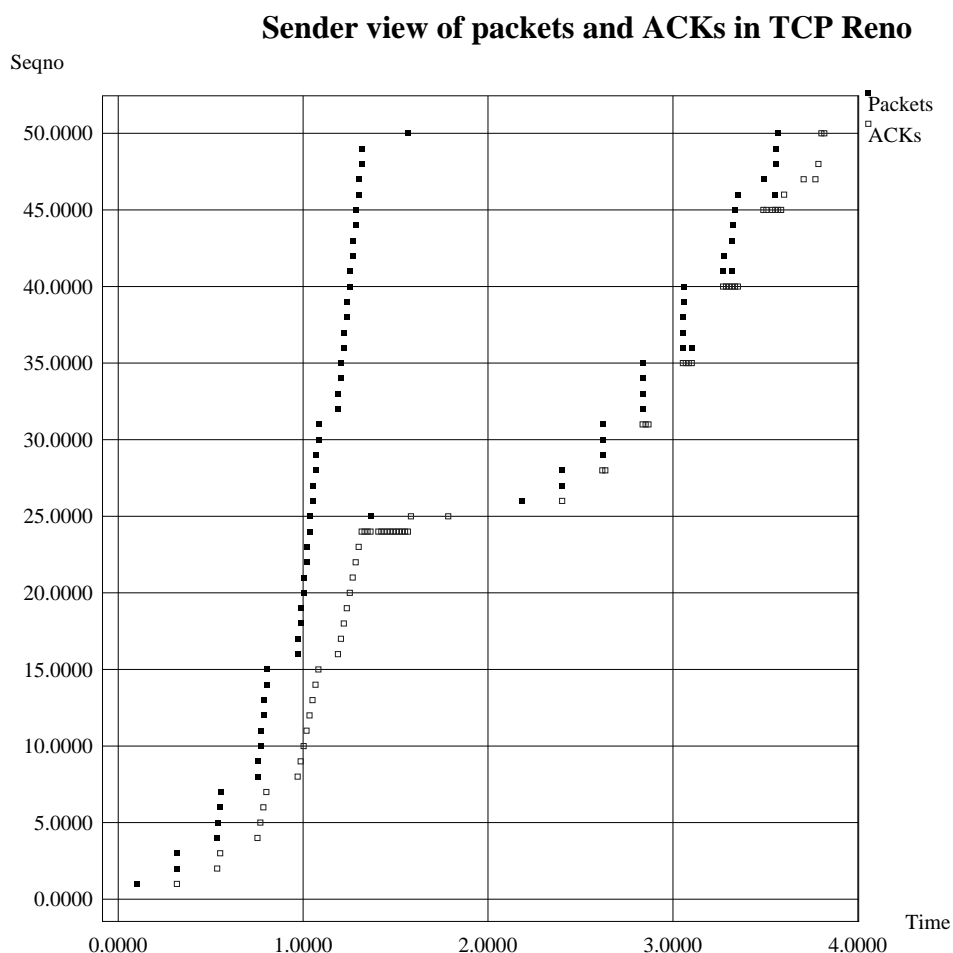


Figure 4: Reno : Fast Retransmission

1. Mark on the x-axis of the graph, the time interval when the connection is in slow-start.

**Answer:** *Slow start is approximately from time 0 to 1.4sec, 2.2 to 3.1 sec, and 3.5 to 3.7 (though it is fine if you missed this last one).*

2. Mark the packets (if any) with × that are lost.

**Answer:** *Packet Numbers: 25, 26, 27, 29, 32, 36, 41, 46, 47, 48, 49*

3. Mark the duplicate packets received by the receiver (if any) with a ○.

**Answer:** *Packet Numbers: 28(2nd), 30(2nd), 31(2nd), 33(2nd), 34(2nd), 35(2nd), 36(3rd), 37(2nd), 38(2nd), 39(2nd), 40(2nd), 41(3rd), 42(2nd), 43(2nd), 44(2nd), 45(2nd), 46(3rd), 50(2nd)*

4. Mark the timed out packet (if any) with a T.

**Answer:** *Packet Number: 26*

5. Draw on the graph an approximate curve showing how a Tahoe TCP Sender would behave.

**Answer:** *Tahoe will behave the same as Reno for the first 1.5sec. Then while Reno would wait for a timeout, Tahoe will start slow start around 1.5sec and continue roughly parallel to Reno but being about 1sec earlier than Reno.*