Introduction to Computer Networks
Homework 3: Solution

Part I

3.15

Consider the cross-country example shown in Figure 3.17. How big would the window size have to be for the channel utilization to be greater than 98 percent? Suppose that the size of a packet is 1,500 bytes, including both header fields and data.

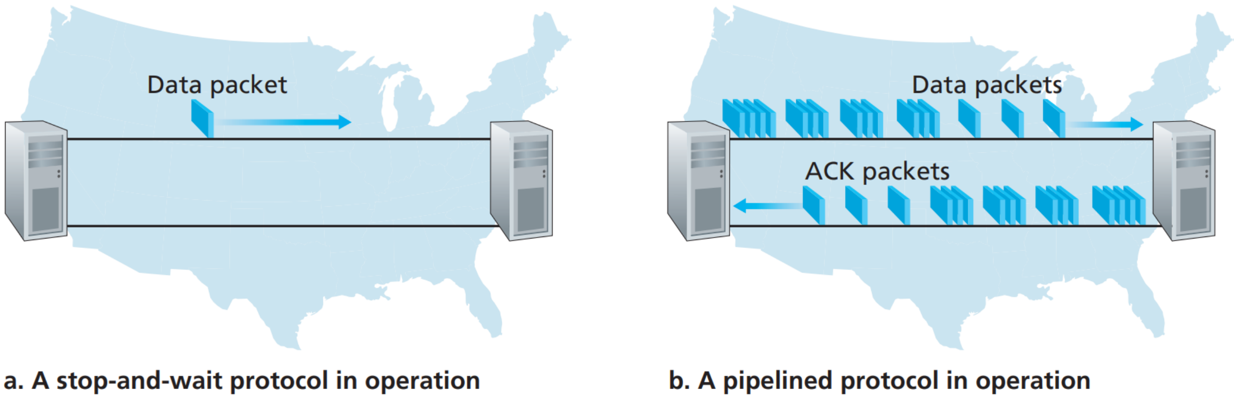


Figure 3.17 ♦ Stop-and-wait versus pipelined protocol

The speed-of-light round-trip propagation delay between these two end systems, *RTT*, is approximately 30 milliseconds. Suppose that they are connected by a channel with a transmission rate, *R*, of 1 Gbps (10^9 bits per second).

$$\frac{L}{R} = \frac{1500\ (bytes) \times 8\ \left(\frac{bits}{byte}\right)}{10^9\ (bits/sec)}$$
$$= 0.000012\ (sec) = 0.012\ (ms)$$

It takes 0.012 milliseconds to send a packet. According to the problem, a required utilization of 0.98 has been specified.

$$\frac{\frac{L}{R}}{\frac{L}{R} + RTT} \times n = 0.98 \qquad \frac{0.012n}{0.012 + 30} = 0.98$$

$$0.012n = 29.41176 \qquad n = 2450.98 \Rightarrow 2451$$

Therefore, the window size would have to be approximately 2451.

3.22

Consider the GBN protocol with a sender window size of 4 and a sequence number range of 1,024. Suppose that at time t, the next in-order packet that the receiver is expecting has a sequence number of k. Assume that the medium does not reorder messages. Answer the following questions:

(a) What are the possible sets of sequence numbers inside the sender's window at time t? Justify your answer.

1. $[k-4, k-3, k-2, k-1]$
   $[k-3, k-2, k-1, k]$
   $[k-2, k-1, k, k+1]$
   $[k-1, k, k+1, k+2]$
   $[k, k+1, k+2, k+3]$

2. Let's consider two boundary cases with a window size $N = 4$.

   (1) Suppose the receiver has received packet $k-1$, and has ACKed that and all other preceding packets. If all these ACKs have been received by sender, then sender's window is $[k, k+N-1]$.

   (2) Suppose next that none of the ACKs have been received at the sender. In this second case, the sender's window contains $k-1$ and the N packets up to and including $k-1$. The sender's window is thus $[k-N, k-1]$.

(b) What are all possible values of the ACK field in all possible messages currently propagating back to the sender at time t? Justify your answer.

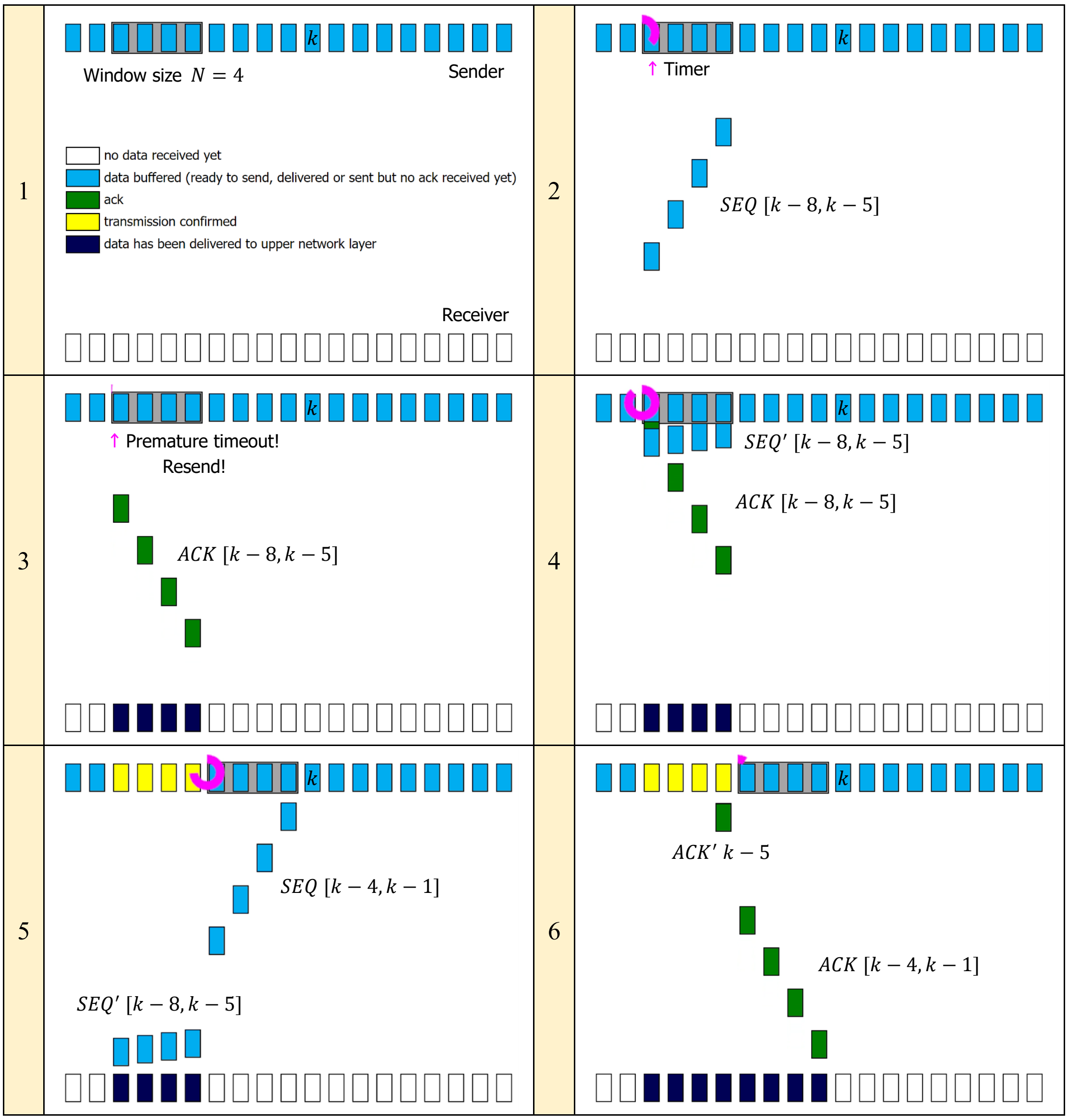
1. $k-5,\ k-4,\ k-3,\ k-2,\ k-1$

2. Let's consider two cases:

   (1) If the receiver is waiting for packet k, then it has received (and ACKed) packet $k-1$ and the $N-1$ packets before that. If none of those N ACKs have been yet received by the sender, then ACK messages with values of $[k-N, k-1]$ may still be propagating back.

(2) The second scenario occurs during a ***premature timeout***, where the sender has already transmitted packet $k-8$ to $k-5$. However, due to network congestion or a too short timeout period, it results in an early timeout, leading to the retransmission of the $k-8$ to $k-5$ packets. Following this, the transmission of packets $k-4$ to $k-1$ begins.

As the receiver has already received packets $k-8$ to $k-5$, it will send ACK for $k-5$. Subsequently, it will sequentially send ACKs for $k-4$ to $k-1$. Therefore, the possible values for ACK range from $k-5$ to $k-1 \implies [k-N-1, k-1]$



3.27

Host A and B are communicating over a TCP connection, and Host B has already received from A all bytes up through byte 126. Suppose Host A then sends two segments to Host B back-to-back. The first and second segments contain 80 and 40 bytes of data, respectively. In the first segment, the sequence number is 127, the source port number is 302, and the destination port number is 80. Host B sends an acknowledgment whenever it receives a segment from Host A.

(a) In the second segment sent from Host A to B, what are the sequence number, source port number, and destination port number?

In the second segment from Host A to B, the sequence number is 207, source port number is 302 and destination port number is 80.

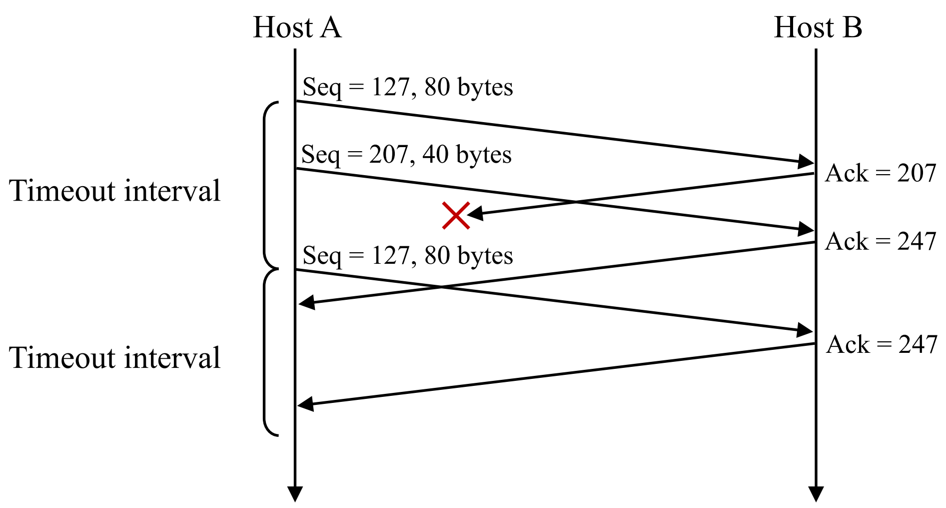
(b) If the first segment arrives before the second segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number, the source port number, and the destination port number?

If the first segment arrives before the second, in the acknowledgement of the first arriving segment, the acknowledgement number is 207, the source port number is 80 and the destination port number is 302.

(c) If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number?

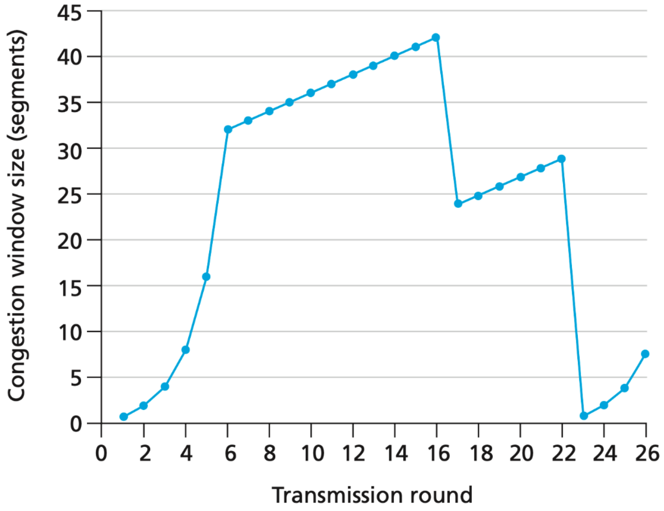
If the second segment arrives before the first segment, in the acknowledgement of the first arriving segment, the acknowledgement number is 127, indicating that it is still waiting for bytes 127 and onwards.

(d) Suppose the two segments sent by A arrive in order at B. The first acknowledgment is lost and the second acknowledgment arrives after the first timeout interval. Draw a timing diagram, showing these segments and all other segments and acknowledgments sent. (Assume there is no additional packet loss.) For each segment in your figure, provide the sequence number and the number of bytes of data; for each acknowledgment that you add, provide the acknowledgment number.



3.40

Consider Figure 3.58. Assuming TCP Reno is the protocol experiencing the behavior shown above, answer the following questions. In all cases, you should provide a short discussion justifying your answer.



(a) Identify the intervals of time when TCP slow start is operating.

TCP slow start is operating in the intervals [1, 6] and [23, 26]

(b) Identify the intervals of time when TCP congestion avoidance is operating.

TCP congestion avoidance is operating in the intervals [6, 16] and [17, 22]

(c) After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

After the 16th transmission round, packet loss is recognized by a triple duplicate ACK.
If there was a timeout, the congestion window size would have dropped to 1.

(d) After the 22nd transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

After the 22nd transmission round, segment loss is detected due to timeout, and hence the congestion window size is set to 1.

(e) What is the initial value of *ssthresh* at the first transmission round?

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

(f) What is the value of *ssthresh* at the 18th transmission round?

The threshold is set to half the value of the congestion window when packet loss is detected. When loss is detected during transmission round 16, the congestion windows size is 42. Hence the threshold is 21 during the 18th transmission round.

(g) What is the value of *ssthresh* at the 24th transmission round?

The threshold is set to half the value of the congestion window when packet loss is detected. When loss is detected during transmission round 22, the congestion windows size is 29. Hence the threshold is 14 (taking lower floor of 14.5) during the 24th transmission round.

(h) During what transmission round is the 70th segment sent?

During the 1st transmission round, packet 1 is sent; packet 2 ~ 3 are sent in the 2nd transmission round; packets 4 ~ 7 are sent in the 3rd transmission round; packets 8 ~ 15 are sent in the 4th transmission round; packets 16 ~ 31 are sent in the 5th transmission round; packets 32 ~ 63 are sent in the 6th transmission round; packets 64 ~ 96 are sent in the 7th transmission round. Thus packet 70 is sent in the 7th transmission round.

(i) Assuming a packet loss is detected after the 26th round by the receipt of a triple duplicate ACK, what will be the values of the congestion window size and of *ssthresh*?

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 the new threshold value + 3 MSS. Thus, the new values of the congestion window size and threshold will be 7 and 4 respectively.

(j) Suppose TCP Tahoe is used (instead of TCP Reno), and assume that triple duplicate ACKs are received at the 16th round. What are the *ssthresh* and the congestion window size at the 19th round?

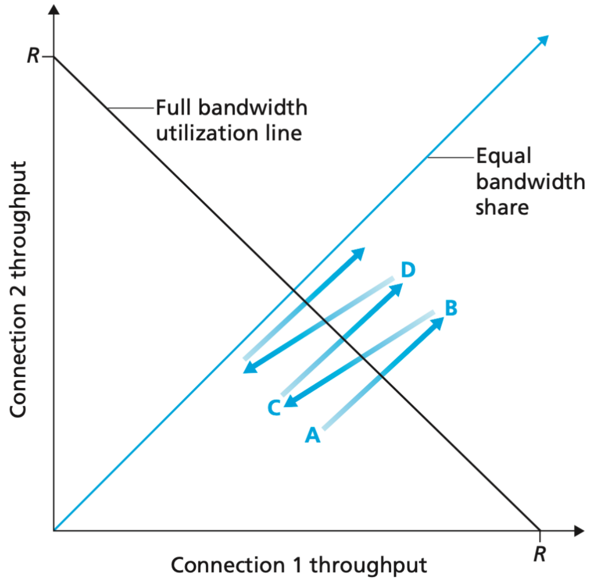
The threshold is 21, and the congestion window size is 4. Since triple duplicate ACKs are detected at the 16th round, the slow start occurs at the 17th round, and the threshold is 42 ÷ 2 = 21, also, the congestion window size is set to 1, hence the congestion window size is 4 at the 19th round.

(k) Again suppose TCP Tahoe is used, and there is a timeout event at 22nd round. How many packets have been sent out from 17th round till 22nd round, inclusive?

17th: 1 packet; 18th: 2 packets; 19th: 4 packets; 20th: 8 packets; 21st: 16 packets;
22nd: 21 (threshold) packets. So, the total number is 1 + 2 + 4 + 8 + 16 + 21 = 52.

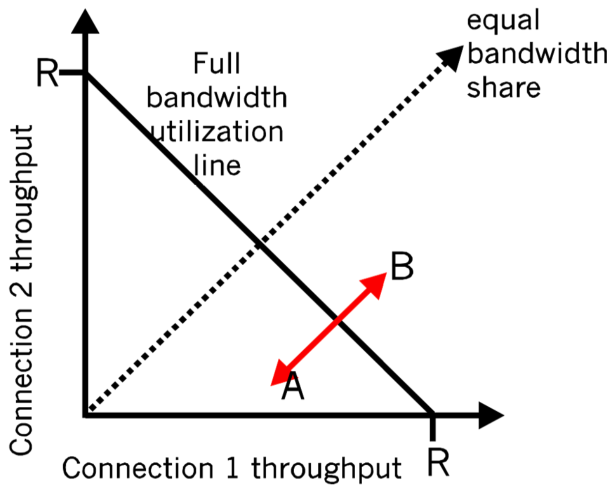
3.41

Refer to Figure 3.55, which illustrates the convergence of TCP's AIMD algorithm. 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 diagram similar to Figure 3.55.

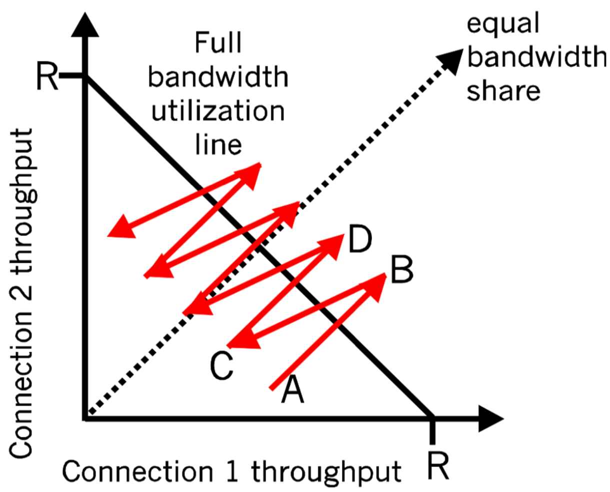


Refer to the figures below. In Figure (a), the ratio of the linear decrease on loss between connection 1 and connection 2 is the same - as ratio of the linear increases: unity. In this case, the throughputs never move off of the AB line segment.

In Figure (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.



(a) linear increase, with equal linear decrease



(b) linear increase, connection 1 decrease is twice that of connection 2

3.43

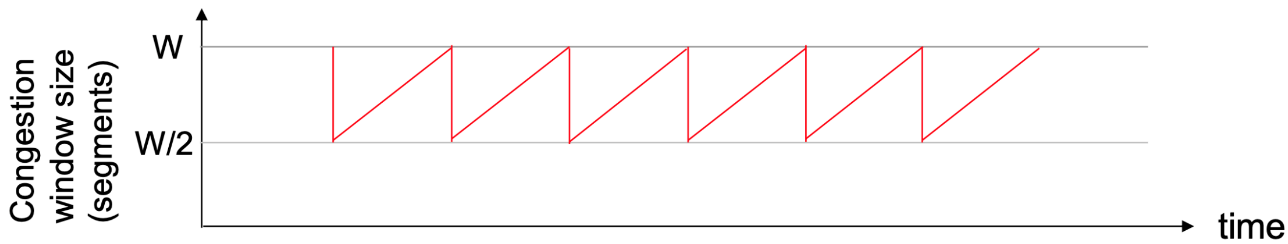
Host A is sending an enormous file to Host B over a TCP connection. Over this connection there is never any packet loss and the timers never expire. Denote the transmission rate of the link connecting Host A to the Internet by R bps. Suppose that the process in Host A is capable of sending data into its TCP socket at a rate S bps, where $S = 10 \cdot R$. Further suppose that the TCP receive buffer is large enough to hold the entire file, and the send buffer can hold only one percent of the file. What would prevent the process in Host A from continuously passing data to its TCP socket at rate S bps? TCP flow control? TCP congestion control? Or something else? Elaborate.

In this problem, there is no danger in overflowing the receiver since the receiver's receive buffer can hold the entire file. Also, because there is no loss and acknowledgements are returned before timers expire, TCP congestion control does not throttle the sender. However, the process in host A will not continuously pass data to the socket because the send buffer will quickly fill up. Once the send buffer becomes full, the process will pass data at an average rate or $R << S$.

Part II. Additional problems.

II.1

Consider TCP's *congestion avoidance* phase in which the congestion window size is additive-increase and multiplicative-decrease. Suppose that the length of each segment is identical (which is equal to 1 MSS) and the maximum available bandwidth for a sender before congestion happens is equal to W segments all the time. Assume that the round-trip time (denoted by RTT) for the sender is a constant all the time. In this situation, the congestion window size over time is shown as the red curve in the figure below. Assuming $W >> 1$, what is the average throughput (in bits per second)? Write down your solution in terms of W, MSS, and RTT, where the unit of MSS is bits.



Consider the period of time when the congestion window size varies from $\frac{W}{2}$ to W.

The average throughput should be

$$\frac{\frac{\left(\frac{W}{2}+W\right)\cdot period}{2}\cdot MSS}{period \cdot RTT} = \frac{3W \cdot MSS}{4 \cdot RTT}$$

II.2

Recall the macroscopic description of TCP throughput. In the period of time from when the connection's rate varies from $W/(2 \cdot RTT)$ to W/RTT, only one packet is lost (at the very end of the period). Hint: Use the figure in Problem II.1.

(a) Show that the loss rate (fraction of packets lost) is equal to

$$\boldsymbol{L = loss\ rate = \frac{1}{\frac{3}{8}W^2 + \frac{3}{4}W}}$$

By II.1, consider the period of time when the congestion window size varies from $\frac{W}{2}$ to W.

By the definition of AIMD, the congestion window size increase by one $cwnd$ for each RTT given window size is not the maximum. Thus, total package transmitted in a period would be

$$\frac{W}{2} + \left(\frac{W}{2} + 1\right) + \cdots + W$$

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

$$= \frac{W^2}{4} + \frac{W}{2} + \frac{\left(\frac{W}{2} + 0\right)\left(\frac{W}{2} + 1\right)}{2}$$

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

$$= \frac{3W^2}{8} + \frac{3W}{4} \quad \text{(The number of segments transmitted)}$$

Total package transmitted in one period is

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

Therefore, the loss rate is

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

(b) Use the result above to show that if a connection has loss rate L, then its average rate is approximately given by

$$\approx \frac{\mathbf{1.22 \cdot MSS}}{\boldsymbol{RTT\sqrt{L}}}$$

For W is large,

$$\frac{3W^2}{8} >> \frac{3W}{4}$$

We can derive that

$$W = \sqrt{\frac{8}{3L}}$$

Thus, the average throughput is

$$\frac{3W \cdot MSS}{4 \cdot RTT} = \frac{3 \cdot MSS}{4 \cdot RTT}\sqrt{\frac{8}{3L}} = \frac{1.22 \cdot MSS}{RTT \cdot \sqrt{L}}$$