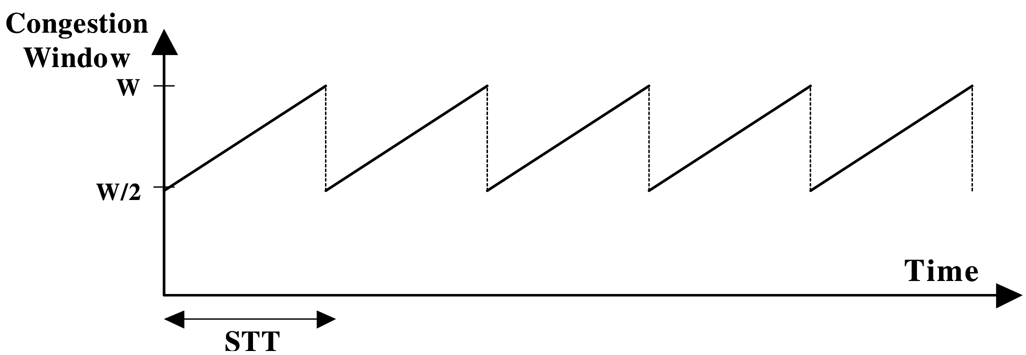
# Homework 3

*Due: 12 Nov 2015, 11:59pm*

## Problem 1




The picture above shows the famous TCP saw tooth behavior. We are assuming that fast retransmit and fast recovery always work, i.e. there are no timeouts and there is exactly one packet lost at the end of each "tooth". We are assuming that the flow control window is large and that the sender always has data to send, i.e. throughput will be determined by TCP congestion control.

In the picture, W represents the congestion window size at which a congestion packet loss occurs (expressed in maximum transfer units). You can assume that W is large, so feel free to approximate (W-1) or (W+1) by W. STT represents the "saw tooth time" expressed in seconds.

a. *Calculate the average throughput T for this connection as a function of the roundtrip time (RTT), the maximum transfer unit size (MTU), and packet loss rate PLR for this connection. Please use the notation suggested by the figure, i.e. W and STT, as intermediate values if you need them. Explain each step of your answer.* Hint: Calculate STT and amount of data sent in one STT separately and then divide.

   The average window size is 3W/4 (we know that window growth is linear, so it's the average of W and W/2), and a window of data is sent each RTT.

$$Rate = \frac{3W * MTU}{4 * RTT}$$

However, we wish to eliminate W, so we first realize that

$$STT = \frac{W * RTT}{2}$$

because our window grows by one per RTT, and must get from W/2 to W. Next we realize that we lose one packet every STT, so Loss Rate, PLR, is 1 out of every "Sawtooth Packets" = STP

$$STP = STT * PacketRate$$

$$STP = STT\frac{3W}{4 * RTT}$$

$$STP = \frac{W * RTT}{2}\frac{3W}{4 * RTT}$$

$$STP = \frac{3}{8}W^2$$

$$PLR = \frac{8}{3W^2}$$

solving for W,

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

and substitute back into our datarate

$$Rate = \frac{3\sqrt{\frac{8}{3PLR}} * MTU}{4 * RTT}$$

$$Rate = 0.75\sqrt{\frac{8}{3}}\frac{MTU}{RTT * \sqrt{PLR}}$$

$$Rate \approx \frac{1.22MTU}{RTT * \sqrt{PLR}}$$

b. *The above TCP connection is used in an idle network, and the link capacity of the bottleneck link is B. What throughput will the user observe? Please explain briefly.*

$$0.75B$$

is clear from the picture if you assume that the loss event happens very soon after the congestion window results in a send rate of B. I *strongly* encourage you to think about whether that must be true though.

c. *What packet loss rate will the connection suffer? Explain.*

$$0.75B \approx \frac{1.22MTU}{RTT * \sqrt{PLR}}$$

$$0.5625B^2 \approx \frac{1.22^2 MTU^2}{RTT^2 * PLR}$$

$$PLR \approx \frac{2.67MTU^2}{RTT^2 * B^2}$$

# Problem 2

The window-scale option of TCP provides a means for supporting a large advertised window. PAWS provides a means for extending the 32-bit sequence number of TCP.

a. *Suppose we have a one-gigabit/second network running cross-country. It has a one-way delay of 60ms. What is the maximum throughput we can attain if we use 16 bits for the window size? Explain. You may assuming whatever you'd like for other parameters, such as MTU.*

The maximum-size segment we can send is the maximum window size, 64k bytes in this case. After sending such a segment, the send window drops to zero and nothing more can be sent until we receive a window update. If the receiver consumes the data instantly and sends back an ack for all the data along with a window update for another maximum window, the sender can send another maximum-size segment after waiting roundtrip-time seconds after sending the first. In this case, since the roundtrip time is 120 milliseconds, the maximum throughput is 64k-bytes/120 milliseconds, or 546 kilobytes/second (or 4.3 megabits/second).

b. *With the window-scale option we can have windows up to* $2^{30}$*-1 in length (to keep things simple, let's say the maximum window size is 1 gigabyte). With a gigabyte window, what's the maximum throughput we can attain over the network of part 1?*

   One gigabit/second. With a gigabyte window, we are no longer constrained by window size and can, in principle, send one gigabyte every roundtrip time. However, we can't transfer data any faster than the network's bandwidth.

c. *With 32-bit sequence numbers and no PAWS, what problems will we have if we have sustained maximum-speed transfers using the parameters of part 2? Explain, showing an example.*

   With 32-bit sequence numbers and one gigabit/second transmission, wrap-around will happen in around 32 seconds. If a segment starting with sequence number k is sent but is somehow delayed, it might still be active up to the end of the maximum segment lifetime of two minutes. If four gigabytes of data are sent successfully in the meantime and our lost segment reappears, since its sequence number is what's now expected, the receiver might accept it rather than the proper segment with sequence number $k$.

## Problem 3

Suppose we have a ten-gigabit/second network with a roundtrip time of 100 milliseconds. The maximum segment size (MSS) is 1500 bytes.

a. *How long will it take for a TCP connection to reach maximum speed after starting in slow-start mode?*

   The transmission rate starts at one MSS-size segment per roundtrip time, or 15,000 bytes/second. The maximum speed is 1.25 gigabytes/second. Thus it must increase its speed by a factor of 83,333.33. Assuming no packet loss, after each roundtrip time (100 milliseconds), the speed doubles. So, after log2(83,333.33) (= $\approx$16.347) roundtrips, the speed will reach the maximum. Thus maximum speed is reached in 1.7 seconds.

A common mistake was to try to reach a 10 gigabit window size, but you only need a tenth of that because of the 100 ms (1/10 second) RTT.

b. *Assume there is some probability that a packet is lost in transmission, even though there is no congestion. Give a scenario in which the congestion window becomes 1 MSS (i.e., 1500 bytes) and TCP begins growing the window linearly rather than exponentially (i.e., it's not in slow-start mode).*

Suppose the first segment sent is received and ack'd, but the second segment sent is lost. Thus after timing out on the ack for the second segment, the sending TCP sets its congestion window to one MSS and begins the linear increase of its congestion window.

c. *How long will it take the TCP connection to achieve maximum speed after the scenario of part 2, assuming no further packet loss?*

83,333 roundtrips, or 8,333 seconds (two hours, eighteen minutes, and 53 seconds).

d. *Suppose (exactly) every 100th packet is dropped. Approximate the throughput of TCP on this network. How about every 1000th packet?*

## Problem 4

*We have learned that TCP provides a "reliable" byte stream. Consider a simple client/server pair. The client has 1 kilobyte of data it wishes to send to the server. It calls* `connect`*,* `write`*, and* `close` *appropriately, and each returns without error. Explain how:*

a. *The server might not receive the data.*

The data may have been delivered to the server OS (and acknowledged, and the TCP FIN sequence occured), but not yet given to the application when the server machine crashes.

b. *The server might receive corrupted data.*

A segment could become corrupted in a way that does not trigger the TCP checksum to fail. Usually, the link layer would catch such an error, but the link-layer was not specified. Perhaps it is a link-layer with poor checking. Or perhaps an error could be constructed that triggers neither the TCP checksum nor (for example) the Ethernet checksum.