CSEE-5110: Problem Set 6, FS 2015

1. Host A and B are communicating over a TCP connection, and Host B has already received from A all bytes up through byte 248. Suppose Host A then sends two segments to Host B back-to-back. The first and second segments contain 40 and 50 bytes of data, respectively. In the first segment, the sequence number is 249, the source port number is 503, and the destination port number is 80. Host B sends an acknowledgement whenever it receives a segment from Host A.

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

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

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

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

2. Consider SYN cookies.

   (a) Why is it necessary for the server to use a special initial sequence number in the SYNACK?

   (b) Suppose that an attacker knows that a target Host uses SYN cookies. Can the attacker create half-open or fully open connections by simply sending an ACK packet to the target? Why or why not?

3. Consider the TCP procedure for estimating RTT. Suppose that $\alpha = 0.15$. Let `SampleRTT1` be the most recent sample RTT, let `SampleRTT2` be the next most recent sample RTT, and so on.

   (a) For a given TCP connection, suppose 4 acknowledgements have been returned with corresponding sample RTTs `SampleRTT4`, `SampleRTT3`, `SampleRTT2`, and `SampleRTT1`. Express `EstimatedRTT` in terms of the four sample RTTs.

   (b) Generalize your formula for n sample RTTs.

   (c) For the formula you just derived above let n approach infinity. Comment on why this averaging procedure is called an exponential moving average.

4. Why do you think TCP avoids measuring the `SampleRTT` for retransmitted segments?

5. After a timeout event, the timeout period is doubled. This is a form of congestion control. Why does TCP need a window based congestion-control mechanism in addition to this doubling-timeout-interval mechanism?

6. Recall the macroscopic description of TCP throughput. In the period of time from when the connection's rate varies from $W/(2\cdot\text{RTT})$ 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

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

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

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

7. In this problem we investigate whether either UDP of TCP provides a degree of end-point authentication.

   (a) Consider a server that receives a request within a UDP packet and responds to that request within a UDP packet (for example, as done by a DNS server). If a client with IP address X spoofs its address with address Y, where will the server send its response?

   (b) Suppose a server receives a SYN with IP source address Y, and after responding with a SYNACK, receives an ACK with IP source address Y with the correct acknowledgement number. Assuming the server chooses a random initial sequence number and there is no "man in the middle," can the server be certain that the client is indeed at Y (and not at some other address X that is spoofing Y)?

8. Consider sending a large file from a host to another over a TCP connection that has no loss.

   (a) Suppose TCP uses AIMD for its congestion control without slow start. Assuming `cwnd` increases by 1 MSS every time a batch of ACKs is received and assuming approximately constant round-trip times, how long does it take for `cwnd` to increase from 5 MSS to 11 MSS (assuming no loss events)?

   (b) What is the average throughput (in terms of MSS and RTT) for this connection up through time = 6 RTT?

9. When closing a TCP connection, why is the two-segment-lifetime timeout not necessary on the transition from LAST-ACK to CLOSED?

10. You are hired to design a reliable byte-stream protocol that uses a sliding window (Like TCP). This protocol will run over a 1-Gbps network. The RTT of the network is 100 ms, and the maximum segment lifetime is 30 seconds.

    (a) How many bits would you advertise in the `AdvertisedWindow` and `SequenceNum` fields of your protocol header?

    (b) How would you determine the numbers given above, and which values might be less certain?

11. If host A receives two SYN packets from the same port from remote host B, the second may be either a retransmission of the original or, if B has crashed and rebooted, an entirely new connection request.

    (a) Describe the difference as seen by host A between these two cases.

    (b) Give an algorithmic description of what the TCP layer needs to do upon receiving a SYN packet. Consider the duplicate/new cases above and the possibility nothing is listening to the destination port.

12. Suppose a TCP connection, with window size 1, loses every other packet. Those that do arrive have RTT = 1 second. What happens? What happens to `TimeOut`? Do this for two cases:

    (a) After a packet is eventually received, we pick up where we left off, resuming with `EstimatedRTT` initialized to its pre-timeout value, and `TimeOut` to double that.

    (b) After a packet is eventually received, we resume with `Timeout` initialized to the last exponentially backed-off value used for the timeout interval.

13. Suppose TCP's measured RTT is 1.0 second except that every Nth RTT is 4.0 seconds. What it the largest N, approximately, that doesn't result in timeouts in the steady state (i.e. for which the Jackobson/Karels `TimeOut` remains greater than 4.0 seconds)? Use $\delta = 1/8$.

14. Consult *Request for Comments* 793 to find out how TCP is supposed to respond if a FIN or an RST arrives with a sequence number other than `NextByteExpected`. Consider both when the sequence number is within the receive window and when it is not.