**CSCE 560 Homework / Wireshark Lab 3**

**Chapter 3 – Transport Layer**

**Fall 18**

**Assigned: Monday, 22 Oct**

**Due: Wednesday, 14 Nov, 1400**

**Problem 1**. Chapter 3, R3

Consider a TCP connection between Host A and Host B. Suppose that the TCP segments traveling from Host A to Host B have source port number x and destination port number y. What are the source and destination port numbers for the segments traveling from Host B to Host A?

**Source port will be y and the destination port will be x.**

**Problem 2**. Chapter 3, R4

Describe why an application developer may choose to run an application over UDP rather than TCP. [Provide at least three reasons.]

**The application can control exactly what data is sent and when. UDP does this through passing data to the network layer immediately because there is no congestion control. There is no additional delay for setting up a connection. It’s a very simple protocol with no connection state at the sender or receiver. There are no buffers and few variables to maintain. The server can support more UDP clients than TCP clients.**

**Problem 3**. Chapter 3, R6

Is it possible for an application to enjoy reliable data transfer even when the application runs over UDP? If so how?

**Yes, it is possible for an application to enjoy reliable data transfer even when the application runs over UDP. This can be done if reliability is built into the application itself. For example, by adding acknowledgement and retransmission mechanisms to the application. Google Chrome implements reliability in an application-layer protocol on top of UDP.**

**Problem 4**. Chapter 3, R8

Suppose that a Web server runs in Host C on port 80. Suppose this Web server uses persistent connections, and is currently receiving requests from two different hosts: A and B. Are all of the requests being sent through the same socket at Host C? If they are being passed through different sockets, do both of the sockets have port 80? Discuss and explain.

**The requests from A and B are sent through different sockets. The sockets are identified by the source port, source IP address, destination port, and destination IP address. Since A and B have different IP addresses they will be assigned two different sockets. When A and B send segments to the server, *all* segments will have**

**destination port 80. In particular, both the initial connection-establishment segments and the segments carrying HTTP request messages will have destination port 80. As we have just described, the server distinguishes the segments from the different clients using source IP addresses and source port.**

**Problem 5.** Chapter 3, R14

True of False:

1. Host A is sending Host B a large file over a TCP connection. Assume Host B has no data to send to Host A. Host B will not send acknowledgements to Host A because Host B cannot piggyback the acknowledgements on data.

**False, TCP requires acknowledgements for segments sent over the connection**

1. The size of the TCP RcvWindow (rwnd) never changes throughout the duration of the connection.

**False, rwn=RcvBuffer-[LastByteRcvd-LastByteRead]**

1. Suppose Host A is sending Host B a large file over a TCP connection. The number of unacknowledged bytes that A sends cannot exceed the size of the receive buffer.

**True**

1. Suppose Host A is sending a large file to Host B over a TCP connection. If the sequence number for a segment of this connection is m, then the sequence number for the subsequent segment will necessarily be m + 1.

**False, depends on the size of the data being transmitted**

1. The TCP segment has a field in its header for RcvWindow (rwnd).

**True**

1. Suppose that the last SampleRTT in a TCP connection is equal to 1 sec. The current value of TimeoutInterval for the connection will necessarily be >= 1 sec.

**False, TimeoutInterval = EstimatedRTT + 4((1-B)DevRTT+B\*|SampleRTT-EstimatedRTT|)**

1. Suppose Host A sends Host B one segment with sequence number 38 and 4 bytes of data. Then in this same segment the acknowledgement number is necessarily 42.

**False, that would necessarily be the acknowledgement number in the segment sent in response by Host B. Host A’s acknowledgement depends on the byte stream number concerning the data A has received from B.**

**Problem 6**. Chapter 3, R15

Suppose Host A sends two segments back-to-back to Host B over a TCP connection. The first segment has sequence number 90; the second has sequence number 110.

a. How much data is in the first segment?

**20 bytes**

b. Suppose that the first segment is lost, but the second segment arrives at B. In the acknowledgment that Host B sends to Host A, what will be the acknowledgment number?

**The acknowledgement number will be 111**

**Problem 7**. Chapter 3, R16

Consider the Telnet example discussed in Section 3.5. A few seconds after the user types the letter ‘C’ the user types the letter ‘R’. After typing the letter ‘R’ how many segments are sent and what is put in the sequence number and acknowledgment fields of the segments?

**There would be three more segments. Host A would send Seq=44, Ack=81, data=’R’. Host B would respond with Seq=81, Ack=45, data=’R’. Host A would respond to this with Seq=45, Ack=80.**

**Problem 8**. Chapter 3, P15 [Modified problem from text]

Consider the cross-country example shown in Figure 3.17. How big would the window size (number of segments) have to be for the channel utilization to be greater than 95 percent? Suppose that the size of a packet is 1,500 bytes, including both header fields and data.

**Problem 9**. Chapter 3, P26

Consider transferring an enormous file of L bytes from Host A to Host B. Assume an MSS of 536 bytes.

a. What is the maximum value of L such that TCP sequence numbers are not exhausted? Recall that the TCP sequence number field has four bytes.

**bytes**

b. For the L you obtain in (a), find how long it takes to transmit the file. Assume that a total of 66 bytes of transport, network, and data-link header are added to each segment before the resulting packet is sent out over a 155 Mbps link. Ignore flow control and congestion control, so A can pump out the segments back-to-back and continuously.

**numSegments=L/536=8012998 full segments and one segment of 368 bytes**

**newL=8012998(536+66) + 368+66=4823825230 bytes**

**time=(newL\*8 bits)/155e6 bps = 248.9716 seconds**

**Problem 10**. Chapter 3, P40

Consider the following plot of TCP window size as a function of time. Assuming TCP Reno is the protocol experiencing the behavior shown, answer the following questions. In all cases, you should provide a short discussion justifying your answer.

1. Identify the intervals of time when TCP slow start is operating.

**0-6**

1. Identify the intervals of time when TCP congestion avoidance is operating.

**6-16**

1. After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

**Triple duplicate**

1. After the 22th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

**timeout**

1. What is the initial value of Threshold (ssthresh) at the first transmission round?

**32**

1. What is the value of Threshold (ssthresh) at the 18th transmission round?

**21**

1. What is the value of Threshold (ssthresh) at the 24th transmission round?

**13**

1. During what transmission round is the 70th segment sent?

**7**

1. 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 Threshold (ssthresh)?

**4**

**Problem 11**. Chapter 3, Supplemental Question 1

Visit http://www.iana.org. What are the well-known port numbers for the simple file transfer protocol (SFTP)? For the network news transfer protocol (NNTP)?

**SFTP: uses TCP port 22**

**NNTP: uses TCP port 119**

**Wireshark Lab**

Complete the lab in 03 - Wireshark\_TCP.pdf.

You will notice several packets labeled as **[TCP segment of a reassembled PDU]**. This can be very confusing for someone new to Wireshark, so I recommend disabling this feature in Wireshark by unchecking **Allow subdissector to reassemble TCP streams** in Edit 🡪 Preferences 🡪 Protocols 🡪 TCP.

Combine your answers to questions 7 and 8 in a table similar to the following.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sequence # | Time sent | Time ACK rcd | RTT  (seconds) | Est RTT (seconds) | Seg Length (bytes) |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |