**2nd Lt David Crow**

**ENG/20M**

**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?

The source port number is y. The destination port number is 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.]

An application developer may choose to utilize UDP instead of TCP for one of the following three reasons (among others):

1. UDP can be faster because it doesn’t require a preliminary handshake to establish a connection.
2. UDP allows a developer to choose exactly when data is sent. TCP’s congestion-control mechanism may limit data transfer in an undesirable way. UDP has no such limit.
3. UDP segments have smaller packet headers than their TCP counterparts. This can lead to fewer packets for the same amount of data.

**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. The application’s developer(s) can build reliable data transfer into the application-layer protocol.

**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.

Because this web server uses persistent connections, it first creates a connection for host A and another connection for host B. Additionally, because a socket is uniquely identified by four pieces of information (destination and source IP addresses; destination and source port numbers), these two connections pass through different sockets.

Because host A and host B do not possess the same IP address *and* port number, both sockets can have the same destination IP address and host number (that is, port 80 on host C). In other words, both sockets connect to port 80 on host C and are still uniquely identified by their source IP address and port number.

**Problem 5**. Chapter 3, R14

True or 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. Piggybacking is an optimization technique, not a requirement, and TCP will always send acknowledgements. In other words, Host B will still send acknowledgements even when it can’t piggyback data.

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

False. The rwnd, by definition, indicates the amount of spare room in the buffer. Because the space utilized in the buffer changes over time, the size of the rwnd also changes over time.

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. If the segment with sequence number has bytes, the segment after sequence will have a sequence number of . Of course, isn’t necessarily equal to .

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. According to the equations in the book, depends on and (which is usually ). Thus, isn’t necessarily greater than or equal to 1 second.

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. The ACK number in ’s segment depends on the next expected sequence number from , not on ’s current sequence number or amount of data.

**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?

There are bytes in the first segment.

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 90 because the receiver is still expecting a segment that contains byte 90.

**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?

Three segments are sent. The first segment comes from the client; it contains a sequence number of 43 and an acknowledgement number of 80 (it also contains the letter ‘R’). The second segment comes from the server; it contains a sequence number of 80 and an acknowledgement number of 44 (it contains the server’s echo of the letter ‘R’). The third segment comes from the client and serves only to ACK the server’s segment; this segment contains a sequence number of 44 and an acknowledgement number of 81.

**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.

Round-trip time:

Transmission rate:

Transmission delay:

Utilization:

Number of segments:

Thus, the window size must be about 2376 segments.

**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.

1. 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.

There are possible sequence numbers. Because each sequence number can represent (as few as) one byte, we can represent bytes before we run out of sequence numbers. Thus, the maximum value of is gigabytes with the available sequence numbers.

1. 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.

We require segments to transmit bytes. This means we also need to transmit bytes of headers. In total, we need to transmit bytes.

This will require seconds to transmit the file.

**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.

TCP slow start runs over the transmission round intervals [1, 6) and [23, 26).

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

Congestion avoidance runs over the round intervals [6, 16) and [17, 22).

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

Segment loss is detected by a triple duplicate ACK; we know this because the window size drops to one-half of the window size before the loss.

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

Segment loss is detected by a timeout; we know this because the window size drops to exactly 1 segment.

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

At first, ; we know this because, at , congestion avoidance takes over.

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

At this point, ; we know this because, after the triple duplicate ACK, the window size dropped to 21.

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

At this point, ; we know this because, after the timeout, the threshold was set to one-half of the congestion window size at the time of timeout; at that point, the congestion window size was 26.

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

The 70th segment is sent in transmission round 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)?

Upon the receipt of a triple duplicate ACK, is halved and the congestion window size is set to the new threshold. If a triple duplicate ACK occurs after round 26, both and the congestion window size will be 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 port 115. NNTP uses 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.

[redacted]

Okay. This combined table is shown in Problem 8 of my Wireshark lab submission.