**Assignment #3**

***Introduction to Networks and Communications***

***CS 455/555***

***Department of Computer Science***

***Old Dominion University***

**1-** 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.

**Answer:**

A connection socket is identified with following four fields:

* Source IP address
* Source port number
* Destination IP address
* Destination port number

The Web server creates a separate **connection socket** for each persistent connection. The requests from A and B pass through different sockets. When host C receives an IP datagram, it examines these four fields in the datagram/segment to determine to which socket it should pass the payload of the TCP segment. The identifier for both of these sockets has 80 for the destination port but the identifiers for these sockets have different values for source IP addresses. Unlike UDP, when the transport layer passes a TCP segment’s payload to the application process, it does not specify the source IP address, as this is implicitly specified by the socket identifier.

**2-** Suppose a process in Host C has a UDP socket with port number 6789. Suppose both Host A and Host B, each send a UDP segment to Host C with the destination port number 6789. Will both of these segments be directed to the same socket at Host C? Explain your answer.

**Answer:**

Yes, both of these segments will be directed to the same socket at Host C. The operating system will provide the process with the IP addresses to determine the origins of the individual segments, for each received segment, at the socket interface.

**3-** - Suppose Host A sends two TCP 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?

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?

**Answer:**

a) The first segment contains 110 − 90 = **20 bytes** of data.

b) Since TCP acknowledgments are cumulative, host B will acknowledge that it has received everything up to and excluding sequence number **90**.

**4-** Describe why an application developer might choose to run an application over UDP rather than TCP.

**Answer:**

An application developer may choose to run an application over UDP and may not want its application to use TCP’s congestion control, which can throttle the application’s sending rate at times of congestion. Designers of IP telephony and IP videoconference applications, often choose to run their applications over UDP because they want to avoid TCP’s congestion control. Also, some applications do not need the reliable data transfer provided by TCP.

**5-** UDP and TCP use 1s complement for their checksums. Suppose you have the following three 8-bit bytes: 01010011, 01100110, 01110100. What is the 1s complement of the sum of these 8-bit bytes? (Note that although UDP and TCP use 16- bit words in computing the checksum, for this problem you are being asked to consider 8- bit sums.) Show all work. Why is it that UDP takes the 1’s complement of the sum; that is, why not just use the sum? With the 1s complement scheme, how does the receiver detect errors? Is it possible that a 1-bit error will go undetected? How about a 2- bit error?

**Answer:**

Wrap if there is overflow

0 1 0 1 0 0 1 1

+ 0 1 1 0 0 1 1 0

1 0 1 1 1 0 0 1

1 0 1 1 1 0 0 1

+ 0 1 1 1 0 1 0 0

0 0 1 0 1 1 1 0

1’s complement is 1 1 0 1 0 0 0 1

To detect errors, the receiver adds the four words (the three original words and the checksum). If the sum contains a zero, the receiver knows there has been an error. All one-bit errors will be detected, but two-bit errors can be undetected (e.g., if the last digit of the first word is converted to a 0 and the last digit of the second word is converted to a 1)

**6-** Consider the figure below in which TCP a sender and receiver communicate over a connection in which the sender-to-receiver segments may be lost. The TCP sender sends initial window of five segments at t=1,2,3,4,5, respectively. Suppose the initial value of the sender-to-receiver sequence number is 126 and the first five segments each contain 522 bytes. The delay between the sender and the receiver is 7 time units, and so the first segment arrives at the receiver at t=8. As shown in the figure, two of the five segment(s) are lost between the sender and the receiver.

Answer the following questions and fill in the table:

a) Give the sequence numbers associated with each of the five segments sent by the sender

b) List the sequence of acknowledgements transmitted by the TCP receiver in response to the receipt of the segments actually received. In particular, give the value in the acknowledgement field of each receiver-to-sender acknowledgement, and give a brief explanation as to why that particular acknowledgement number value is being used

**Answer:**

1. Segment 1-> sequence number 126 (range 126 to 647)
2. Segment 2->sequence number 648(range 648 to 1168)
3. Segment 3->sequence number 1168(range 1168 to 1688)
4. Segment 4->sequence number 1688(range 1688 to 2206)
5. Segment 5->sequence number 2206(range 2206 to 2726)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender-toReceiver | Time segment sent | Sender-to-receiver segment sequence number field value | Time segment received, and ACK segment sent | Receiver-to-sender ACK field value |
| Segment 1 | 1 | 126 | 8 | 648 |
| Segment 2 | 2 | 648 | 9 | 1168 |
| Segment 3 | 3 | 1168 | Never received | 10 |
| Segment 4 | 4 | 1688 | Never received | 11 |
| Segment 5 | 5 | 2206 | 12 | 2726 |

**7-** Suppose that TCP's current estimated values for the round trip time (estimatedRTT) and deviation in the RTT (DevRTT) are 260 msec and 15 msec, respectively (see Section 3.5.3 for a discussion of these variables). Suppose that the next measured values of the RTT is 310. Compute TCP's new value of estimatedRTT, DevRTT, and the TCP timeout value after each of these three measured RTT values is obtained. Use the values of α = 0.125 and β = 0.25.

**Answer:**

EstimatedRTT = (1- α)\*EstimatedRTT + α\*SampleRTT

DevRTT = (1-β)\*DevRTT + β\*|SampleRTT-EstimatedRTT|

TimeoutInterval = EstimatedRTT + 4\*DevRTT

Given:

EstimatedRTT=260msec DevRTT=15msec

α=0.125 β=0.25

SampleRTT=310

**Calculation:**

EstimatedRTT = (1-0.125)\*260\*10-3+0.125\*310=(227.5+38.75)msec=266.25msec

DevRTT = (1-0.25)\*15\*10-3+0.25\*abs(310-266.25)=11.25msec+43.75msec=55msec

TimeoutInterval = 266.25msec + 4\*55=486.55 msecs

**8-** Consider the cross-country example shown in Figure 3.17 from the textbook. 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.

**Solution:**

R =109 bps

RTT = 30 msec

Size of packet = 1500 bytes = 12000 bits

Transmission delay of a single data frame = = L / R = 12000 / = 0.012 msec

Channel utilization = (N) (dtrans ) / (RTT +drans ), where N is the given window size

0.98 = (N) (0.012) / (30 + 0.012)

Window size = N = ceil ((0.98) (30.012) / (0.012)) ≃ 2451.

**9-** Consider that only a single TCP (Reno) connection uses one 10Mbps link which does not buffer any data. Suppose that this link is the only congested link between the sending and receiving hosts. Assume that the TCP sender has a huge file to send to the receiver, and the receiver’s receive buffer is much larger than the congestion window. We also make the following assumptions: each TCP segment size is 1,500 bytes; the two- way propagation delay of this connection is 150 msec; and this TCP connection is always in congestion avoidance phase, that is, ignore slow start. a) What is the maximum window size (in segments) that this TCP connection can achieve? b) What is the average window size (in segments) and average throughput (in bps) of this TCP connection? c) How long would it take for this TCP connection to reach its maximum window again after recovering from a packet loss?

**Answer:**

a) Let W denote the max window size measured in segments.

Then, W\*MSS/RTT = 15Mbps, as packets will be dropped if the maximum sending rate exceeds link capacity.

Thus, we have W\*1200\*8/0.16=15\*10^6, then W is about 250.

b) As congestion window size varies from W/2 to W, then the average window size is 0.75W=187.5 segments.

Average throughput is 187.5\*1200\*8/0.16=11.25Mbps.

c) (250/2) \*0.16= 20 seconds, as the number of RTTs (these TCP connections needs in order to increase its window size from W/2 to W) is given by W/2. (Recall the window size increases by one in each RTT.)

**10-** Consider the following figure. Assuming TCP Reno is the used protocol, 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.

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

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

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

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

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

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

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

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?

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?

**Answer:**

1. [1, 6] and [23, 26] are the intervals of time when TCP slow start is operating
2. [6, 16] and [17, 22] are the intervals of time when TCP congestion avoidance is operating
3. Segment loss is detected by a triple duplicate ACK
4. Segment loss is detected by a timeout
5. 32 is the initial threshold at the first transmission round
6. 42/2=21 ,21 is the value of threshold at the 18Th transmission round
7. 26/2=13, 13 is the value of threshold at the 24Th transmission round
8. At 6th round, it has sent 1+2+4+8+16+32=63 segments

At 7th round, it sends 33 segments from 64, including 70th segment

1. Size of threshold =4

Congestion Window size=4

1. Threshold=42/2=21

Congestion window size=1\*2\*2=4