

Homework 3 – Transport Layer

Assigned: Thursday, October 15, 2009

Due: Tuesday, October 27, 2009 *at the beginning of class*

100 points

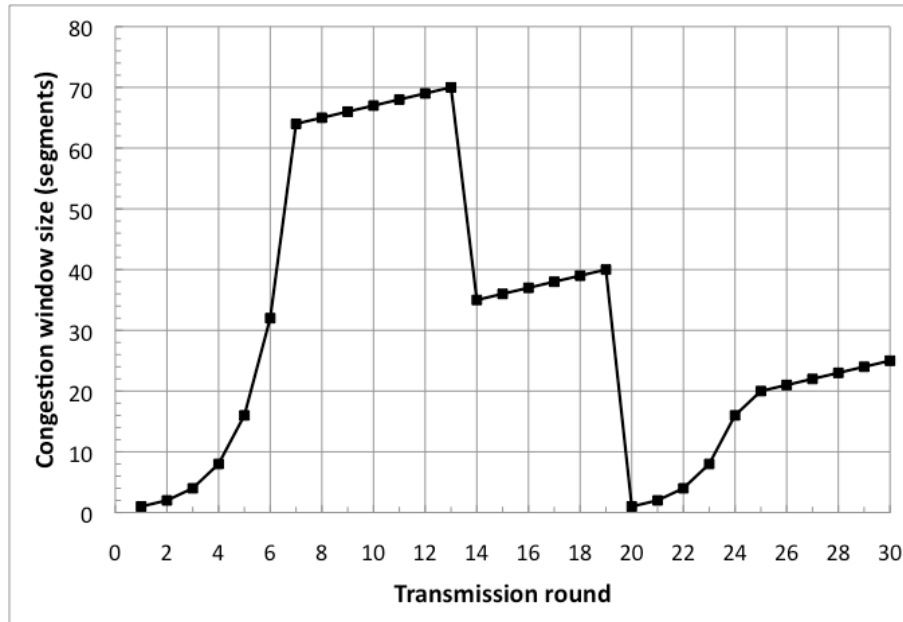
Note: All homework assignments should be done on your own, and your answers should be in your own words. The textbook and lecture notes may be used, but you should not copy verbatim from either of them. *Use of previous years' assignments/solutions or the textbook's solutions manual is **not** permitted.*

Review Questions [48 points, 3 pts each]

1. What is the term for a transport-layer packet? a network-layer packet? a link-layer packet?
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? Why or why not?
3. What does it mean for a reliable transfer protocol to be stop-and-wait?
4. In a reliable transfer protocol, can a sender tell the difference between a lost data packet and a lost ACK? If not, why not? If so, how?
5. How many sequence numbers are needed in a pipelined reliable transfer protocol to avoid ambiguity when the window size is w ?
6. What is the size of the TCP header without options?
7. What is the MSS?
8. Why shouldn't we set the TCP timeout value to be extremely large to avoid early timeouts?
9. How is the receiver's advertised window used by TCP?
10. Why does TCP need to exchange three packets (instead of just exchanging two, for instance) during connection setup (3-way handshake)?
11. What are two symptoms of congestion?
12. During TCP congestion avoidance, how is the congestion window incremented? Answer the question for each ACK and for each RTT.
13. In TCP Tahoe, what happens to the congestion window and the slow-start threshold upon detecting a packet loss?
14. What is fast recovery?
15. **[6 pts]** According to the article "You Don't Know Jack About Network Performance" (link is on schedule under Oct 15 and on Useful Links page), key factors that affect a network application's performance, besides bandwidth, are "network latency, transport protocol buffer management, congestion control dynamics, and the design of the application's protocol". Explain how congestion control dynamics can affect a network application's performance.

Problems [37 points]

1. Host A is sending a 20,000-byte file to Host B using a sliding window protocol. Packets are limited to 1000 bytes each, packets are numbered by packet number starting at 1, and the window size is 6 packets. Packet 10 is lost.
 - a. **[3 pts]** Which packets are retransmitted if Host A and Host B are using the Go-Back-N protocol?
 - b. **[3 pts]** Which packets are retransmitted if Host A and Host B are using the Selective Repeat protocol?
2. **[3 pts]** Draw a diagram of the packets exchanged in TCP connection setup between Host A and Host B. Include TCP flags, sequence numbers, and ACK numbers as appropriate. Assume that Host A's initial sequence number is 1234 and Host B's initial sequence number is 5555.
3. Consider that a TCP sender, Host A, wants to send 50,000 bytes to Host B. The maximum segment size (MSS) is 1460 bytes, the initial congestion window is 1 MSS, and the initial slow-start threshold is 500 MB.
 - a. **[3 pts]** How many TCP segments will it take to send the 50,000 bytes?
 - b. **[3 pts]** How many RTT rounds will it take to send the 50,000 bytes, ignoring connection setup? (Consider that in one 'RTT round', an entire window's worth of packets is sent.)
4. **[4 pts]** Consider that a TCP sender, Host A, wants to send a 30,000,000-bit MP3 file to Host B. The slowest link between Host A and Host B is 2.5 Mbps, the RTT (including transmission delays) is 150 ms, and the TCP MSS is 11,680 bits. The operating system of Host B sets a default receiver window of 16 segments. Assume that there is no congestion in the network. Will Host B be able to receive the file at 2.5 Mbps? Explain why or why not. If not, what would an appropriate receiver window size be?
5. Consider the following plot of the TCP Reno window size as a function of the transmission round (think RTT).



- [3 pts]** Identify the intervals of time when TCP slow-start is operating. How can you tell?
- [3 pts]** Identify the intervals of time when TCP congestion avoidance is operating. How can you tell?
- [3 pts]** After the 13th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout? How can you tell?
- [3 pts]** After the 19th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout? How can you tell?
- [3 pts]** What is the initial value of the slow-start threshold at the first transmission round? How can you tell?
- [3 pts]** What is the value of the slow-start threshold at the 20th transmission round? How can you tell?

Design (15 points)

In Program 4 (for CS 555 only), you are going to implement a sender and receiver that use an alternating bit reliable transport protocol. **In this assignment (for both CS 455 and CS 555),** you are going to outline the implementation.

Recall that in alternating bit reliable transport protocols, there are only two sequence numbers needed, 0 and 1. Each time the receiver gets a message from the sender, it will respond with an ACK. The sender cannot send the next message until the previous one has been acknowledged by the receiver. If an ACK is not received before the timer expires, the previous message must be re-sent.

You should use the RDT 2.2 receiver and the RDT 3.0 sender state machines as described in the lecture notes and in Kurose/Ross as a guide.

Your outline should be typed (not handwritten). You must have an outline for the sender and an outline for the receiver. Each outline must include the variables and methods needed to

implement the protocol. The methods should contain pseudocode describing what functions they perform and what (if any) values they return.

Here's an example for the TCP Ping Client from Program 2:

```
constant:  NUM_PINGS

main()
{
    process command-line arguments to obtain the server and port

    tcpPing (server, port)
}

String createPingMsg (seqno, sendTime)
{
    create a String with the format "PING seqno sendTime"

    return the String
}

long calcRTT (sendTime)
{
    get the current time

    calculate the RTT (current time - sendTime)

    return the RTT
}

tcpPing (server, port)
{
    setup the TCP socket and associated streams

    for (seqno = 0; seqno<NUM_PINGS; seqno++) {

        create a message to send to the server
        1) get the current time as sendTime
        2) createPingMsg (seqno, sendTime)

        send the message to the server

        print the message to the screen

        read the reply from the server

        rtt = calcRTT(sendTime)

        print the rtt
    }

    close the socket
}
```