CSEE 4119: Computer Networks — Due on Tuesday, March 12th, 2013

Assignment #3 – Transport Layer

Srirakumar Balasubramanian, Zheng Cang, Li Yan (TAs) A. Chaintreau (instructor)

How to read this assignment : Exercise levels are indicated as follows

($\rightarrowtail$) "elementary": the answer is not strictly speaking obvious, but it fits in a single sentence, and it is an immediate application of results covered in the lectures.
Use them as a checkpoint: it is strongly advised to go back to your notes if the answer to one of these questions does not come to you in a few minutes.

($\curvearrowright$) "intermediary": The answer to this question is not an immediate translation of results covered in class, it can be deduced from them with a reasonable effort.
Use them as a practice: how far are you from the answer? Do you still feel uncomfortable with some of the notions? which part could you complete quickly?

($\nrightarrow$) "tortuous": this question either requires an advanced notion, a proof that is long or inventive, or it is still open.
Use them as an inspiration: can you answer any of them? does it bring you to another problem that you can answer or study further? It is recommended to work on this question only AFTER you are done with the rest!

Exercise 1: Pipelining and utilization (3 pt)

Consider a 100-Mbps link of 10-km in length with 5 nanoseconds per meter propagation delay, constant 4,000 bit data packets, and negligible processing and queueing delays at both the sender and receiver. Assume the sender always has data to send, that there are no losses or corruptions, and 1Mbps = 10^6bps.

1. ($\curvearrowright$) If pipelining is used with a window N packets, what is the smallest value of N that ensures at least a 80% utilization of the link.

Exercise 2: Difference of GO-BACK-N and SELECTIVE-REPEAT (4 pt)

In the diagram below, one row corresponds to a time slot during which one packet can be sent. Packet and ACKs propagates so that they are received in the following time slot in the other end. The RTT is hence 2 time slots. We assume that the timeout value is set to 5.5 (*i.e.*, after a packet is transmitted, unless an ACK is received the time out is triggered just before the 6th time slot). We assume Pipelining is used (window size 4, each packet size 1). Sequence numbers starts at 0. ACK indicates the sequence number of the packet they acknowledge. All transmissions are successful and without corruption except:

- The first packet with SSN=1 sent by the sender is lost.
- The first ACK sent by the receiver with SSN=3 is lost.

1. ($\rightarrowtail$) Assuming the Sender and Receiver implements GO-BACK-N, represent on the following diagram, for the fist 11 time slots:
    - The packet and ACK sent and received by each host during each time slot, with their SSN.

- The set of packets currently in the window maintained by each host.

GO-BACK-N

Sender	Receiver
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9

2. ($\rightarrowtail$)Do the same diagram for SELECTIVE REPEAT

SELECTIVE-REPEAT

Sender	Receiver
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9

```
0 1 2 3 4 5 6 7 8 9                    0 1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9                    0 1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9                    0 1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9                    0 1 2 3 4 5 6 7 8 9
```

Exercise 3: Designing a reliability protocol for two receivers (12 pt + 1pt for (↬))

A host wishes to transmit message over a broadcast channel to two receivers. Unfortunately, losses may occur on this channel for any of the two receivers independently, so the sender will have to ensure reliability by receiving some signal from each receiver separately.

More precisely, consider a scenario in which a Host A wants to simultaneously send messages to Hosts B and C. A is connected to B and C via a broadcast channel: a packet sent by A (e.g., in a single unreliable_send() operation) is carried by the channel to both B and C.

We make the following assumption on the channel connecting A, B, and C:

- It can independently lose and corrupt messages from A to B and C (and so, for example, a message sent by A might be correctly received at B but not at C).
- It has a maximum bounded delay of D (i.e., if a message is sent by A, it will either be lost or arrive at B and/or C within D time units).
- any control message (e.g., an ACK or NAK) sent by B or C to A will only be received by A and not any other node, but like others they can be lost or corrupted.
- We do not use any pipelining in this channel.

1. (↷) Design a stop-and-wait-like error-control protocol for reliably transferring a packet from A to B and C, such that A will not get new data from the upper layer until it knows that both B and C have correctly received the current packet. Give a FSM description for A and C (assuming the FSM for B is similar, if it is not similar give the FSM for B as well). Also, give a description of the packet format used.

2. (↬) If we assume that each channel ($A \to B, A \to C, B \to A, C \to A$) may create a loss or a corruption independently from the others with probability p_e, and has the same propagation delay d_p. We assume that the packet has a transmission delay d_t and we neglect the transmission delay of the ACK, compute the utilization of this channel.

Exercise 4: Are ACK needed? (6 pt)

In normal conversation, we generally only ask for clarification is something went wrong. Similarly, can we imagine operating a data reliable transmission protocol only using NACK?

1. (⇀) Using only 2 packets and no packet loss, prove that no scheme can operate using only NACK and no sequence number.

2. ($\curvearrowright$) We now consider designing a scheme using sequence-number and NACK. When will the loss of the n_th packet be detected at the receiver? Can timeout protect against some losses?

3. ($\curvearrowright$) In which context does a NACK-only protocol make the most sense? To help you, consider two applications: one is always on and send data frequently, such as a critical sensing operation. The other application is sporadic and asks for a burst of packets, such as multiple web sessions.

Exercise 5: Additive Increase Multiplicative Decrease (6 pt)

Refer to Figure 3.56 that in section 3.7.1 illustrating the convergence of TCP's additive increase, multiplicative decrease algorithm. Suppose that instead of a multiplicative decrease, TCP decreased the window size "linearly", that is by a number that does not depend on the current window size. We consider only two connections and make the same assumption as in this section.

1. ($\curvearrowright$) Let us first assume that all connection decrease their window by the same amount and have the same RTTs. Would the resulting additive increase additive decrease converge to an equal share algorithm from any initial condition? Justify your answer using a diagram similar to Figure 3.56.

2. ($\curvearrowright$) We now assume that, due to different RTTs etc., the effective throughput decreases linearly by a different amounts for the two connections. What will happen then? Does it depend on the initial condition

Exercise 6: The TCP Throughput formula (8 pt)

Consider the macroscopic description of TCP throughput. Note that, in the period of time in which the connections rate varies from $\frac{W}{2RTT}$ to $\frac{W}{RTT}$ only a single packet is lost (at the very end of the period).

1. ($\curvearrowright$) Show that the loss rate (fraction of packets lost) is equal to $L = \frac{1}{\frac{3}{8}W^2+\frac{3}{4}W}$.

   Hint: Consider the number of packets sent during a period.

2. ($\curvearrowright$) Use the result above to show that if a connection has a small loss rate L, then its average rate is approximately given by $\frac{1.22\times MSS}{RTT\times\sqrt{L}}$.