CSE 310 – Computer Networks
Fall, 2020
Programming Assignment 02 – Implementing a Reliable Transport Protocol

Re-Assigned: Monday, 10/05/2019
Due: **Monday, 10/26/2019, at 11:59 PM**

For the second programming assignment you will be implementing a simple reliable transport protocol. The protocol will be unidirectional. Host A will be sending segments, and Host B will be receiving and acknowledging those segments, but Host B will not be sending any application data of its own. Host A is the sender, Host B is the receiver.

1. First, read section 3.4.1 – "Building a Reliable Data Transfer Protocol" of Kurose and Ross, *Computer Networking: A Top-Down Approach*. You will be implementing the Alternating Bit protocol, which they call `rdt3.0`.
2. Read the directions, descriptions, and suggestions in the attached assignment description from the textbook authors carefully.
3. Examine the files prog2.c and prog2.py carefully. All of your code will be added to one of these files to fill in the function stubs. That is how you will implement `rdt3.0` as a part of this network simulation program.
4. You must implement the following functions in prog2.py or prog2.c, please read their descriptions in the assignment description carefully.
   a. `A_init`
   b. `A_output`
   c. `A_input`
   d. `A_timerinterrupt`
   e. `B_init`
   f. `B_input`
5. Do not implement either `B_output` or `B_timerinterrupt`. Those functions are only required for a bi-directional version of the protocol. Please ignore these two function stubs.
6. Your implementation should include some output to STDOUT describing events that occur and protocol actions that are taken during the simulation. For example, message arrival, packet arrival, timer interrupt, or data corruption detection, etc. should all be reported to STDOUT. In addition, describe the actions your functions take in response. For example, building a packet, sending a packet, retransmitting a message, restarting the timer, etc. Part of your submission will be a trace of your program—we need output for the trace.
7. Try running the simulation program first – to see what it looks like before you start adding your own code.
8. Submit your source code, which should consist of a single file—your modified prog2.py or prog2.c
9. Submit a README that briefly explains your strategy for implementing each of the functions required for `rdt3.0`.
10. Submit a trace of your completed program running to the point where about 10 messages have been sent and correctly acknowledged. Use the following settings: loss probability

of 0.1, corruption probability of 0.3, and a trace level of 2. You can use screenshots, or you can redirect the output to a file. You might want to annotate your trace to indicate where interesting things happened (e.g., recovery from packet loss or packet corruption).

Submission Instructions:

1. Please submit all files in a compressed directory as a zip file.
2. Please name your submission file: CSE310_PA02_LastName_FirstName.zip
3. This is an individual programming assignment. Any collaboration on coding will be considered a violation of academic honesty.