Lab Assignment 2

CSC317: Computer Networks

Submission instructions: Submit a PDF file containing your answers and necessary source files containing your code. All files must be submitted on Moodle.

This problem is designed to help students delve into the principles of reliable data transfer. To get you started, you are given a simulation of the Stop-and-Wait (rdt 3.0) protocol in `Stop_and_Wait_sim.py`. Notice that the simulation parameters are defined as global variables at the beginning of the code.

You are also given an image file named `Cornell.JPG`. You must put the image file in the same directory where you put the simulation code. The simulation contains a sender, a receiver and a channel. As you can expect, the channel connects the sender with the receiver. The sender in the simulation reads the image file, splits the file data into segments, creates packet for each segment, and forwards packets to the channel to send to the receiver. The receiver receives packets from channel, extracts segments from packets, delivers segments to the application layer, and forwards ACKs to the channel. The channel then delivers the ACKs to the sender.

To make the channel realistic, we simulate loss and delay of packets/ACKs in the channel. To this end, the channel drops some packets/ACKs with a certain probability. The channel also delays to deliver some packets/ACKs with a certain probability. You can control the loss and delay probabilities by changing the values assigned to the global variables `lossProb` and `delayProb`, respectively.

The receiver reassembles the segments received and reconstructs the image file. If all the segments were delivered to the application layer in the correct order, the reconstructed image would be an exact copy of the original image. Check both the images to check the correctness of the protocol. The simulation also reports if a segment in receiver does not match with the corresponding segment in sender. Check the simulation code carefully to understand how this simulation is done. Then complete the following tasks.

Task 1. Run the simulation for `delayProb = 0`, `lossProb = 20`, and `seqNSpaceSize = 2`. You should observe that the reconstructed file is an exact copy of the original file. Note that setting the size of the sequence number space to 2 essentially makes the protocol a bit-altering (0 or 1 as sequence/ACK numbers) protocol. Now set `delayProb = 30`, and run the simulation again. You will most likely find that the image reconstructed by the receiver is distorted. In this case, the simulation will report which segments in receiver do not match with the corresponding segments in sender. Examine the generated log file to find out what went wrong. Is it a bug in the simulation code, or a limitation of rdt3.0? Justify your answer with a timing diagram. The timing diagram must show the sender on left and receiver on right. Also, the diagram must show all the packets/ACKs with their sequence numbers, and any loss or delay of packets/ACKs.

Task 2. Increment the size of the sequence number space to 3, 4, 5 etc. Does this solve the problem found in task 1? What is the lowest value of the size of the sequence number space that solves the problem? Explain why this number solves the problem. Correct the simulation according to your findings.

Task 3. Extend the simulation to simulate the Go-Back-N (GBN) protocol. Note that we used 10 simulation ticks for the RTT, and 15 simulation ticks for the timeout period in our simulation of the rdt3.0 protocol. Keep these values unchanged for your simulation of GBN. Given that only one packet can be transmitted by the sender in a simulation tick, what should the ideal window size? Implement your simulation with the window size you think is ideal. Don't forget to change the size of the sequence number space accordingly.

Task 4. Plot total simulation times associated with both the protocols (your corrected rdt3.0 and GBN) for the following values:

1. 0%, 10%, 20% and 30% packet loss probabilities with a fixed packet delay probability of 10%.
2. 0%, 10%, 20% and 30% packet delay probabilities with a fixed packet loss probability of 10%.
3. 64, 256, 512, and 1024 bytes MSS.

Note that, the image reconstructed by the receiver should be an exact copy of the original image for all simulation runs.

Finally, give a brief summary of the simulation results. Explain which protocol works better for which situations according to the simulation results.

Submission: You should submit a separate source files for each protocol. For rdt3.0, you need to submit the corrected simulation (with your modified value for the size of the sequence number space). You should also submit three plots and a summary of the simulation results.