CS 158 Project Assignment

Summer 2022 Prof. Navrati Saxena

Team Project: (2 members per team)

Implement a very basic TCP client and server implemented on two separate machines. The TCP client/server will communicate over the network and exchange data to maintain **TCP Sliding window protocol.** The server will start in passive mode listening for a transmission from the client. The client will then start and contact the server (on a given IP address and port number). The client will pass the server an initial string (e.g.: "network"). On receiving a string from a client, the server should respond with connection setup "success" message.

Now the client will start sending the TCP segments in the sliding window manner. For simplicity, instead of sending actual segments, the client will send only the TCP sequence numbers to the server. You can assume a segment consists of 1024 sequence numbers. On reception of the sequence numbers, the server will respond to the client with the corresponding "ACK". The "ACK" of TCP works in the following manner:

- TCP receiver will send an ACK for the next segment that it is expecting and buffers the ones that it has received so far, even if they're out of order. For example, if the receiver has successfully received Sequence numbers *up to n*, it will respond with ACK, having sequence number *n+1*.
- If segments are lost, the sender will not retransmit every un-acknowledged segment. Instead, it just retransmits the oldest un-acknowledged segment.

On receiving an ACK, the client will adjust its sliding window. The sliding window adjustment is guided by the following rules:

- The TCP sliding window will be initialized to 1 segment.
- On every successful transmission and acknowledgement, the sliding window will be increased by twice, until any segment is lost or until the sliding window reaches a maximum value of 2^{16}.
- After reaching the maximum value, on every successful transmission the sliding window maintains the maximum value.
- On detection of any segment loss, the sliding window reduces to half (1/2) of its previous value.
- After any segment loss, on any successful transmission, the sliding window increases linearly by 1 segment.

The server, on the other hand, will continue receiving the sequence numbers and keep track of any missing sequence number. If there is no missing sequence number, the server will consider all the segments until that sequence number are received.

Note: The program should be executed for 10,000,000 segments and maximum sequence number should be limited to 2^{16}.

Output: The server will keep a count of received segments and missing segments. Calculate good-put (received segments/sent segments) periodically after every 1000 segments received at the server and report the average good-put.

Your project will be judged and graded on the following two parameters:

1. Execution of your project – **6 Points**
   The project will be tested on a few test cases so try to make your connection more like TCP in real world. (HINT: Think of different scenarios that may occur between a client-server connection).
2. Submission on Canvas – **14 Points**
   Your submission should include:
   1. Once code file with server-side code. – **2 Points**
   2. One code file with client-side code. - **2 Points**
   3. One .doc/.pdf file containing screenshots graphs and tables as mentioned in the rubrics below:
   Please refer to the below rubrics for points distribution:

Executable code files *(two files- one for server, one for client)* - 2 Points for each file	
Description	**Points**
Code compiles with no errors	0.5
Code is modular/ divided into functions	0.5
Code is well documented	0.5
Code implements the desired project correctly	0.5
One .doc/.pdf file *(containing screenshots graphs and tables)* -4 Points	
Screenshot of output	
Sender and receiver IP address	1
Number of packets sent and received	1
Goodput	2
Graphs - 4 Points	
TCP sender and receiver window size over time	2
TCP sequence number received over time	2
TCP sequence number dropped over time	2

Make sure every file has name of both the team members on it.