COMP3331 Computer Networks and Applications Assignment Specification

Due date: 11:59pm Friday, 19th October 2018, Early Bird: 12th October 2018

What I am required to do?

- Implement a Simple Transport Protocol (STP) **reliable transport** protocol over the **UDP protocol**
- Most Features that requires implementation in STP are in the textbook: Section **3.5.3, 3.5.4 and 3.5.6**
- Example of some Features: ACK, timeout, sequence number
- **Implement reliable uni-directional data transfer for receiver and sender**
- **Data**: Sender -> Receiver
- **ACK**: Receiver -> Sender
- File Names: sender.py & receiver.py (can create additional files such as header files and name them as whatever)

List of features that needs implementation in Sender and Receiver:

1. Three-way handshake (SYN, SYN + ACK, ACK) - connection establishment. ACK sent by the sender to conclude 3-way handshake (Do Not Include PayLoad) - See Section 3.5.6 of text .
2. Four-segment (FIN, ACK, FIN, ACK) connection termination. Sender initiates connection and Closes connection once the file has successfully been transmitted. - See Section 3.5.6
3. Sender must maintain single-timer for timeout operation. Implement Round Trip Estimation and RTO estimation - See Section 3.5.3. The timeout is not a constant value but given by the formula on page 243 (TimeoutInterval = EstmatedRTT + 4 * DevRTT). **Use the initial value of EstimatedRTT = 500milliseconds and DevRTT = 250 milliseconds**
4. Sender need to **implement all features in Section 3.5.4, except for doubling timeout.** STP must include the simplified TCP sender (Figure 3.33) and fat retransmit (pg 249-251). Also concepts discussed in class **(Sequence numbers, cumulative acknowledgements, timers, buffers, etc. . .)**

5. **Receiver implement features in 3.5.4 (Don't need to follow ACK generation). All packets should immediately acknowledged (Don't need to implement delayed ACKs)**

6. STP is a byte-stream oriented protocol. **Include sequence number and acknowledgement number fields in the STP header for each segment. The meaning of sequence number and acknowledgement number are the same as in TCP**

7. MSS (Maximum segment size) is the maximum number of bytes of data that your STP segment can contain. In other words, MSS counts data ONLY and does NOT includ header. Sender must be able to deal with different valeus of MSS. **The value of MSS will be supplied to Sender as an input argument**.

8. Maximum Window Size (MWS) input argument. MWS is the maximum number of un-acknowledged bytes that the Sender can have at any time. MWS counts ONLY data. Header length should NOT be counted as part of MWS. *Note: TCP doesn't define maximum window size. The TCP MWS determined by the smaller value of received window and the congestion control window. In this assignment, we are not required to implement the flow and instead will be taking in MWS parameters.*

LastByteSent - LastByteAcked <= MWS

9. Because sender and receiver are local, there are no possibilities that packets will be dropped/delayed/corrupted. **So implementing some of Packet Loss and Delay (PLD) Module as part of the Sender program is necessary**

Packet header and MSS

- Include fields that are necessary for segment header
- Use the TCP header as inspiration
- Two given important and must have field (Sequence Number and ACK)
- Need number of flags for connection establishment and teardown
- Data portion must not contain more than MSS
- STP segment format for data transfer and acknowledgement must be the same.
- All information for proper functioning of STP must be provided in STP headers
- Dont use any information from header of UDP datagram that will encapsulate the STP segments