CS4316: Computer Networks
Spring 2020

Instructor: Dr. Deepak Tosh
Due Date: 03/30/2020

Programming Assignment # 1

Disclaimer:

This assignment can be done with **a group of 2 students at max**. If one student decides to implement GBN-protocol, the other students should implement Selective-Repeat.

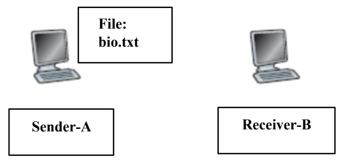
GOAL:

The goal of this assignment is to implement the pipelined reliable data transfer protocol over UDP implemented using the Go-Back-N algorithm and Selective Repeat (SR). We know that GBN uses a sliding window to send more than one packets to a receiver over an unreliable medium where packets can get corrupted and lost too. If a packet or ACK is lost, then the sender retransmits the entire window of packets. The python implementation of GBN and SR should reliably copy a file from the sender's location to receiver's location.

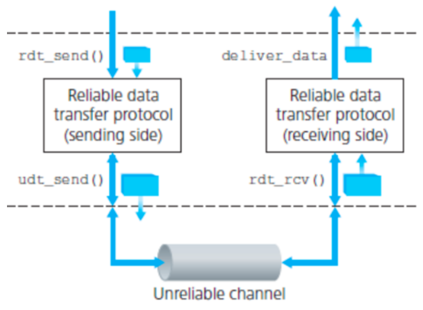
Your code should run on the localhost with the following commands.

- [on Terminal-1]
  Run the receiver: python3 receiver.py [GBN|SR] [filename of the file to write to]
- [on Terminal-2]
  Run the sender: python3 sender.py [GBN|SR] [filename of file to read from]

Problem# 1: Implementing File Transfer via Go-Back-N [60 Pts]



You are given with the starter code "Sender.py" and "Receiver.py". In the programs, I have hardcoded the port numbers and localhost. Therefore, you can test your code in your computer itself by running two different terminals or command prompts.


rdt_send()
deliver_data
Reliable data transfer protocol (sending side)
Reliable data transfer protocol (receiving side)
udt_send()
rdt_rcv()
Unreliable channel

The setup is as follows:

1. The Sender-A has a file (let's say "bio.txt") that has to be delivered to the receiver-B.
2. Sender-A has access to "*rdt_send(filename)*" to send the file
3. Receiver-B receives the data (chunks of file data) from GBN protocol by reading from "rdt_rcv()".

Refer to Figure 3.20 and 3.21 from the book to understand the state diagram of GBN sender and receiver.

Your task is to implement the GBN sender protocol for sender-A and the receiver protocol for Receiver-B.

Implementing GBN-Sender

Task 1.1: Implement ***rdt_send(sock, filename)*** that takes the sender's socket and the file name as argument. Its job is to calculate how many packets to send, and then send packets based on available window size. Ensure you implement the actions when timeout and packet error event happens.

Task 1.2: Implement ***receive(sock)*** that checks if the expected ACKs are being received or not. Based on that the sender can take necessary actions.

(**Note**: you are given with 3-auxiliary helper modules: ***timer*, *packet*, *udt***).

Timer module: Provides *start(), stop(), timeout(), running()* functions

Packet module: Provides *make(seqNo, data), extract(packet), make_empty()*

Udt module: Provides *send(packet, sock, addr), rcv(sock)* to send/recv from unreliable channel

Implementing GBN-Receiver:

Task 1.3: Implement ***rdt_rcv(sock, filename)*** at the receiver which receives the packets and writes them into the file referred with filename. Ensure that if the expected sequence number is not received, it sends the last in-ordered packet's sequence # as ACK.

Problem# 2: Implementing File Transfer via Selective Repeat [40 Pts]

Repeat the same implementation as above, but this time you need to implement the Selective Repeat (SR) protocol instead of the GBN.

Task 2.1: Read the SR-sender's events and actions from Fig 3.24 in the Book and modify the "Sender.py" program accordingly.

Task 2.2: Read the SR-receiver's events and actions from Fig 3.25 in the Book and modify the "Receiver.py" program accordingly.

Submission:

Submit your **Python codes for both algorithms** on **Blackboard only**. Also, your **report** should describe the design of the program functions through comments and ***show the sample output through screenshots***.

- Each student's contribution should be reported with a comparative chart as shown below and should be reported through the report.

Task lists	Student-1's contribution	Student-2's contribution
…	…	…

If you have any questions, comments, concerns, doubts, or confusions, please stop by my office to clarify. You can also email me on dktosh@utep.edu. I will be very happy to help.