Assignment 7

Reliable Transport Protocol (RelTP) using Datagram Socket

R.Sri Charan Reddy 14CS10037
G.Sai Bharath Chandra 14CS10020

Computer Networks Laboratory

FILES SUBMITTED

There are six files in total:

1) *RelTP_sender.c*
2) *RelTP_receiver.c*
3) *UDP_server.c*
4) *UDP_client.c*
5) *Topo.py*
6) *Report*

Design of RelTP

Window size = **4**
Timeout(t) = **3 sec**
Packet size = **500 bytes**

Sender

- The Sender reads the data from the file and stores it in the window.
- The sender sends the packets and waits for the acknowledgement from the receiver
- It sends maximum of 5 packets at once before receiving an acknowledgement
- Whenever an ack packet is received , it slides the packets which were sent and adds new packet to the window
- If no packet is received within the timeout, it resends the packets present in the window

Receiver

- The receiver keeps track of the sequence number of the latest packet received.
- It accepts a packet if the sequence number of the packet is one more than the present
- If not it rejects the packet
- Irrespective of it rejects or accepts the packet , it sends an ack packet containing the present sequence number.

Instructions to run

RelUDP:

Run both RelTP_client.c and REITP_server.c in separate systems using below commands:

“gcc -o s REITP_sender.c”

“./s <Server IP> <Server Port> <Chunk Size> <Window Size> <File Name to Send> ”

“gcc -o r REITP_receiver.c”

“./r <UDP Server Port> <Chunk Size><File Name into which it should be downloaded> ”

UDP:

Run both UDP_server.c and UDP_client.c in separate systems using below commands:

“gcc -o s UDP_server.c”

“./s ”

“gcc -o c UDP_client.c”

“./c ”

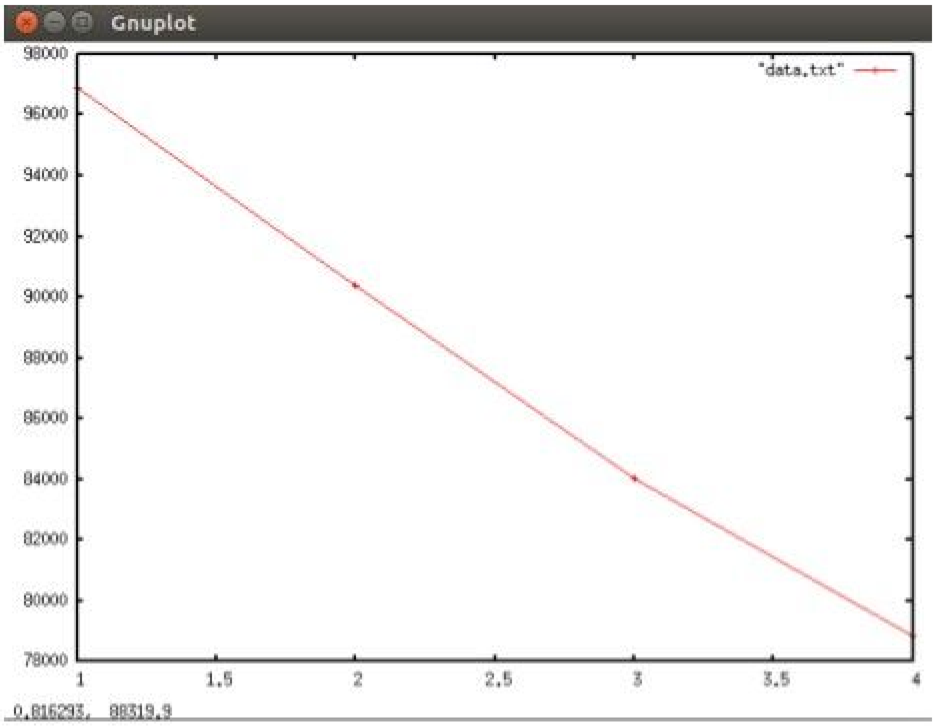
Put the filename you want to transfer and server IP in UDP_client.c

#define SERVER_IP "10.146.5.34" and *strcpy(file,"file_name");*

Observations:

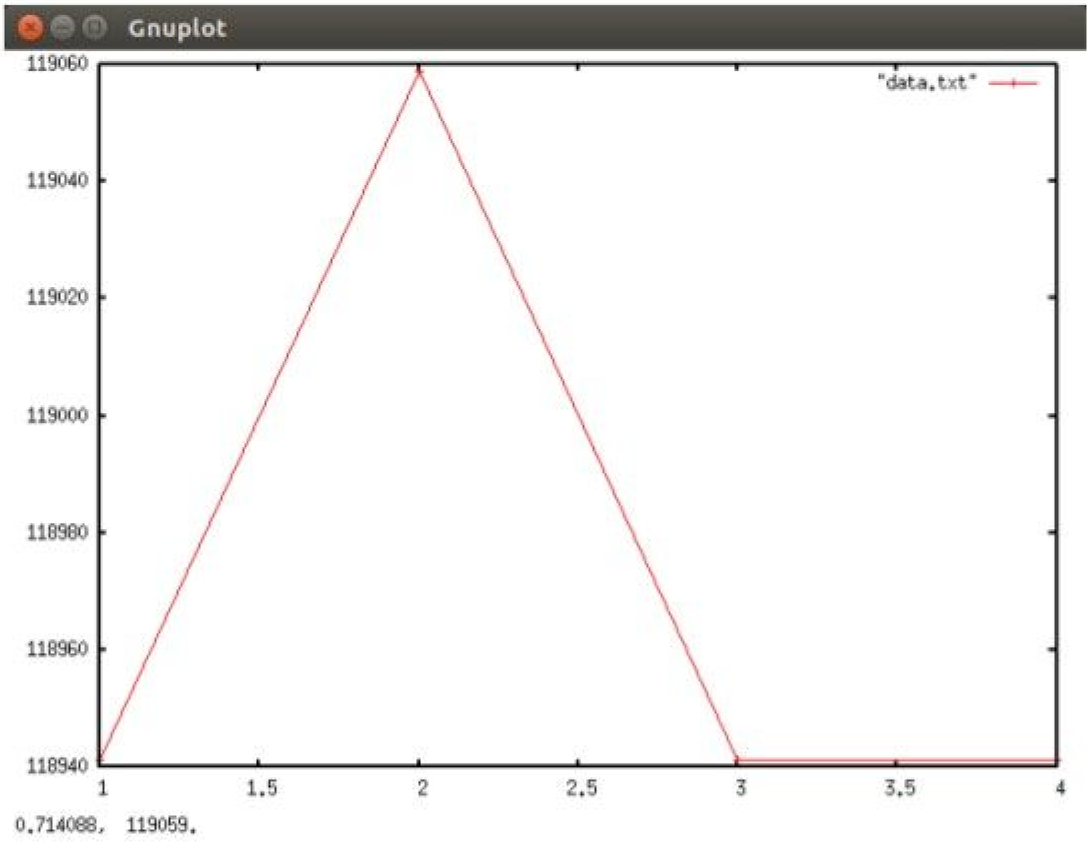
<u>RelUDP average throughput for 1Mbps Bandwidth vs Loss rate at the link:</u>

Loss rate at the link	Average throughput(bits/sec)
1	775122.267936
2	723239.083872
3	672276.209896
4	630672.618848



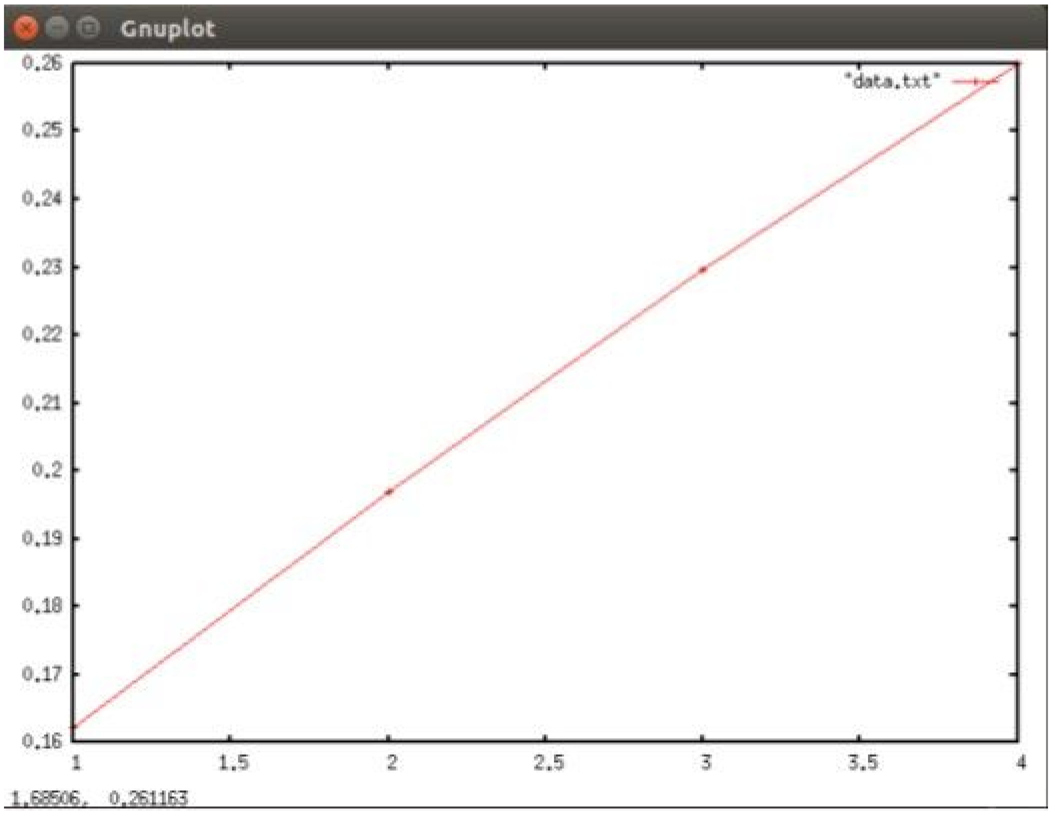
Standard UDP average throughput for 1Mbps Bandwidth vs Loss rate at the link:

Loss rate at the link	Average throughput(bits/sec)
1	951529.411768
2	952470.588232
3	951529.411768
4	951529.411768



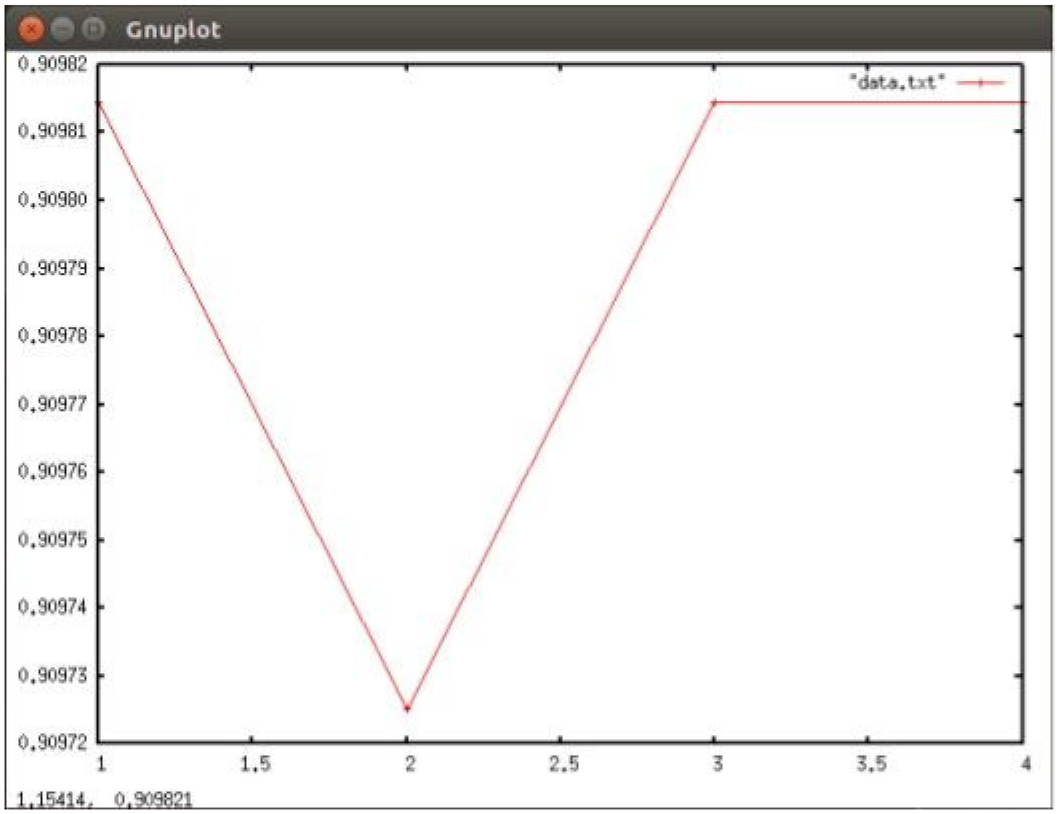
<u>RelUDP average packet loss rate for 1Mbps Bandwidth vs Loss rate at the link:</u>

Loss rate at the link	Average packet loss rate(lost packets/sec)
1	0.16232386015
2	0.19704026596
3	0.22957649243
4	0.25993546782



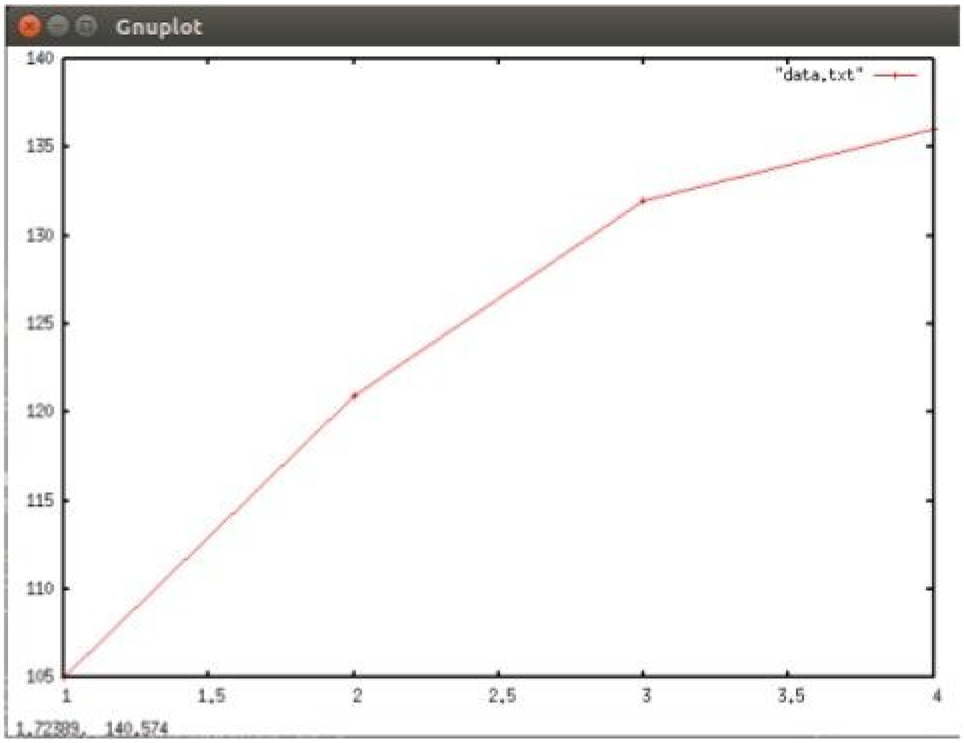
<u>Standard UDP average packet loss rate for 1Mbps Bandwidth vs Loss rate at the link:</u>

Loss rate at the link	Average packet loss rate(lost packets/sec)
1	0.90981432294
2	0.90972511851
3	0.90981432294
4	0.90981432294



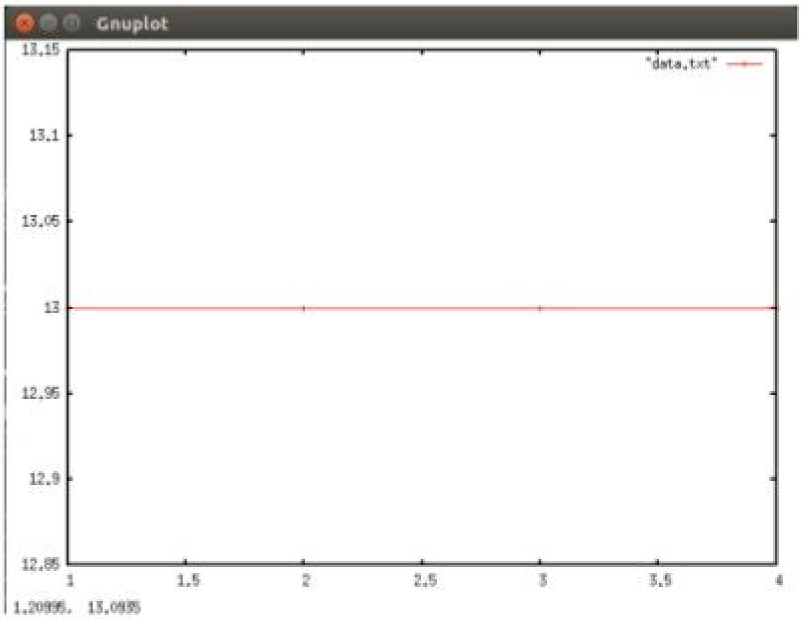
<u>RelUDP File transfer time for 10Mbps Bandwidth vs Loss rate at the link:</u>

Loss rate at the link	File transfer time(min)
1	105.000000
2	121.000000
3	132.000000
4	136.000000



Standard UDP File transfer time for 10Mbps Bandwidth vs Loss rate at the link:

Loss rate at the link	File transfer time(min)
1	13.000000
2	13.000000
3	13.000000
4	13.000000



Justifications:

UDP has better throughput but many packets are lost.(loss percentage is more).It has a better throughput because it has no acknowledgement and doesn’t wait for any packets.

Timeout value, if it is high it takes a lot of time to receive all the packets as we need to wait for acknowledgements of all of them and if it is too low we may not receive all packets,so here optimally it is chosen as ***3 sec***.

Window size, if it is high,and if the packet is lost then it takes a lot of time to send it again and if it is low,it takes more time to receive every acknowledgement of each packet,so here optimally it is chosen as ***4***.

Packet size, if it is high,and if the packet is lost then it takes more time to send it all over again due to it size and if it is low, total number of packets will be high and takes time to wait for acknowledgement of all packets,so here optimally it is chosen as ***500 bytes***.