CENG435 TERM PROJECT PART-1 STUDY REPORT

Yavuz Selim YESILYURT
2259166
Group-68

Gokhan SAN
2171916
Group-68

I. Design and Implementation Part

A. Project Overview

The goal of the project is to design and implement a network consist of a source node, a broker, two routers and a destination node. Between nodes source and broker, TCP (Transmission Control Protocol) is employed. Source sends byte streams to the broker node then the broker packetizes these byte streams and send the packets to the routers over UDP (User Datagram Protocol). Thus between broker and router nodes UDP is employed. Then, between destination and router nodes, UDP will be employed so that routers can store and forward the packets to the destination node. On this constructed network we carry out three experiments regarding finding the average end-to-end delay on different scenarios with many sample executions. The findings about these experiments are plotted to a network emulation delay vs end-to-end delay graph with a 95% confidence interval and discussed in detail, later on this report.

B. Design Approach

Source node was needed to send byte streams to the broker node with TCP. But since we need to calculate end-to-end delay in the destination node we embedded time information to these byte streams in source node beforehand. Then, the broker node's role was to get these byte streams and convert them to the UDP datagrams and send them to the routers. So, we designed the broker node to make needed convertions. Then, we did come up with a mechanism for the broker to share workload over two routers. Mechanism was simply breaking the incoming byte streams to equal-size chunks and packetizing them with embedding sequence numbers to the packets along with the time information that we got from source node's byte streams. Routers roles were simple, they were to just store and forward incoming coming packets to the destination node. In destination node, first combining incoming chunks and then a calculation for end-to-end delay got carried out. Also combined messages are accumulated temporarily to save them to a persistent storage after the experiment.

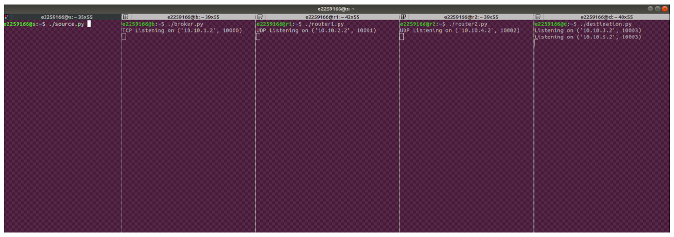
C. Implementation Approach

The implementation process started by us choosing Python language to write a single script for each node. We decided to embed required information to the packets as character bytes since our sensor signals were also in the same form and they would be easy to handle.

Then we created our slices on GENI platform by using given tutorials and XML file that defines our topology with assigned manual IP adresses.

We started the implementation of nodes from Source node. The first thing we implemented was to read some sensor signals from a file and turn them into byte streams. To achieve this goal we created a file with specific byte amount to represent these sensor readings with a bash script. Then we created an IPV4 TCP socket to connect to the broker's ready-to-accept TCP socket. Afterwards source node was to read 1024 bytes from file we created and since the end-to-end delay is measured between source node and destination, it was to add starting time information for each data that is read from file. Upon getting final form of the packet, source node was to send it to the broker as TCP streams over the sockets and the ports we reserved. Since each device has one or two IP adress to store, we didn't choose to create an IP table for routing but we embedded static IP addresses which are needed to communicate for each node to the script as global variables and named them accordingly.

Then we implemented the script for the Broker node. The broker node gets the TCP streams, packetizes them and convert them to UDP datagrams to send two routers. First thing to do was creating one listening TCP socket (from Source) and two outgoing sockets (for routers) for the broker. For listening TCP socket, we changed the socket flag to prevent execution from being port-blocked by the operating system for security reasons and we binded it to the listen address for the source and set it to listen for incoming messages. Upon getting byte streams with TCP socket, broker's task was to breaking it to chunks of size 128 bytes and sending them to destination over routers with sharing the workload. To achieve this, we first extracted time information from the message, started a sequence number from 0 and we



```
e2259166@s: ~
```

```
e2259166@s: ~ 35x55
e2259166@s:~$ ./source.py
```

```
e2259166@b: ~ 39x55
e2259166@b:~$ ./broker.py
TCP Listening on ('10.10.1.2', 10000)
```

```
e2259166@r1: ~ 42x55
e2259166@r1:~$ ./router1.py
UDP Listening on ('10.10.2.2', 10001)
```

```
e2259166@r2: ~ 39x55
e2259166@r2:~$ ./router2.py
UDP Listening on ('10.10.4.2', 10002)
```

```
e2259166@d: ~ 40x55
e2259166@d:~$ ./destination.py
Listening on ('10.10.3.2', 10003)
Listening on ('10.10.5.2', 10003)
```

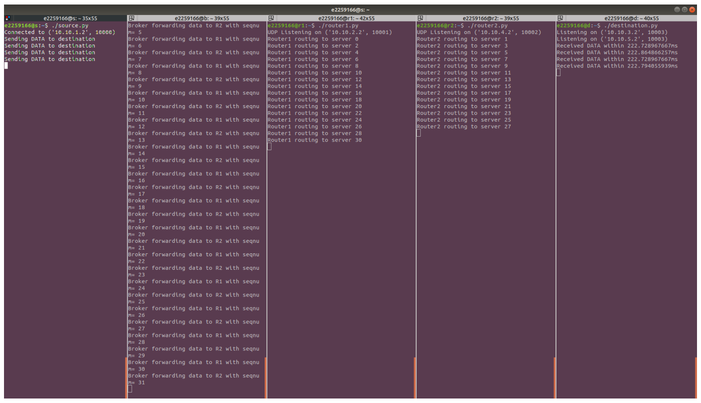
Fig. 1. Start Up vision for execution

broke the remaining part of the message to 128 bytes. For each of these chunks we embedded 12 bytes of information to the head of the packets, namely 4 bytes sequence number of that chunk and 8 bytes time information of that data and sent them to the routers in even-odd fashion. If sequence number was even we sent it to Router1, else we sent it to the Router2. For sending process in broker, IP addresses and pre-reserved ports of routers, which are static, are given to the outgoing UDP sockets.

Later, our goal was to implement routers. Since routers just store and forward the packets they get from broker to the destination, it was needed for them to listen the outgoing UDP sockets we used on broker node. So each router has a listening UDP socket and a port number that is used by it to bind its listening socket. As soon as they receive packets from broker they store and forward incoming packet with their outgoing UDP socket to the destination node. For sending process in routers, IP addresses of two different interfaces of destination and same pre-reserved port, which are static, are given to the outgoing UDP sockets. Also, as mentioned on broker implementation, packets with an even sequence number are sent over router1 and packets with an odd sequence number are sent over router2.

Destination implementation required two listening UDP sockets to listen indefinetely since there are two routers forwarding packets to the destination. Thus, to achieve this goal, we created a multithreaded mechanism, namely we defined a generic handler function to handle incoming packets and created a thread and gave it this function as parameter, then set this thread to handle incoming packets from Router1 with this function's parameters. Main execution's task was to handle incoming messages from Router2. Hence main execution also set to call this generic handler function with its own specific parameters. Handling incoming messages in destination node was simply get chunks of 128 bytes and storing each of them to a container. For calculation of end-to-end delay, upon getting a whole packet of 1024 bytes, execution immediately calculates the end-to-end delay with extracting packet's time information and subtracting it from current time information. Then it also accumulates this in a container, too.

Figure 1 shows a screenshot of an example start-up vision



```
e2259166@s: ~
```

```
e2259166@s: ~ 35x55
e2259166@s:~$ ./source.py
Connected to ('10.10.1.2', 10000)
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
```

```
e2259166@b: ~ 39x55
Broker forwarding data to R2 with seqnu
m= 5
Broker forwarding data to R1 with seqnu
m= 6
Broker forwarding data to R2 with seqnu
m= 7
Broker forwarding data to R1 with seqnu
m= 8
Broker forwarding data to R2 with seqnu
m= 9
Broker forwarding data to R1 with seqnu
m= 10
Broker forwarding data to R2 with seqnu
m= 11
Broker forwarding data to R1 with seqnu
m= 12
Broker forwarding data to R2 with seqnu
m= 13
Broker forwarding data to R1 with seqnu
m= 14
Broker forwarding data to R2 with seqnu
m= 15
Broker forwarding data to R1 with seqnu
m= 16
Broker forwarding data to R2 with seqnu
m= 17
Broker forwarding data to R1 with seqnu
m= 18
Broker forwarding data to R2 with seqnu
m= 19
Broker forwarding data to R1 with seqnu
m= 20
Broker forwarding data to R2 with seqnu
m= 21
Broker forwarding data to R1 with seqnu
m= 22
Broker forwarding data to R2 with seqnu
m= 23
Broker forwarding data to R1 with seqnu
m= 24
Broker forwarding data to R2 with seqnu
m= 25
Broker forwarding data to R1 with seqnu
m= 26
Broker forwarding data to R2 with seqnu
m= 27
Broker forwarding data to R1 with seqnu
m= 28
Broker forwarding data to R2 with seqnu
m= 29
Broker forwarding data to R1 with seqnu
m= 30
Broker forwarding data to R2 with seqnu
m= 31
```

```
e2259166@r1: ~ 42x55
e2259166@r1:~$ ./router1.py
UDP Listening on ('10.10.2.2', 10001)
Router1 routing to server 0
Router1 routing to server 2
Router1 routing to server 4
Router1 routing to server 6
Router1 routing to server 8
Router1 routing to server 10
Router1 routing to server 12
Router1 routing to server 14
Router1 routing to server 16
Router1 routing to server 18
Router1 routing to server 20
Router1 routing to server 22
Router1 routing to server 24
Router1 routing to server 26
Router1 routing to server 28
Router1 routing to server 30
```

```
e2259166@r2: ~ 39x55
e2259166@r2:~$ ./router2.py
UDP Listening on ('10.10.4.2', 10002)
Router2 routing to server 1
Router2 routing to server 3
Router2 routing to server 5
Router2 routing to server 7
Router2 routing to server 9
Router2 routing to server 11
Router2 routing to server 13
Router2 routing to server 15
Router2 routing to server 17
Router2 routing to server 19
Router2 routing to server 21
Router2 routing to server 23
Router2 routing to server 25
Router2 routing to server 27
```

```
e2259166@d: ~ 40x55
e2259166@d:~$ ./destination.py
Listening on ('10.10.3.2', 10003)
Listening on ('10.10.5.2', 10003)
Received DATA within 222.728967667ms
Received DATA within 222.864866257ms
Received DATA within 222.728967667ms
Received DATA within 222.794055939ms
```

Fig. 2. Halfway execution vision

for the process and Figure 2 shows a screenshot of an example halfway execution vision of the process.

II. Methodology and Motivation

A. Methodology

By using "struct" and "time" libraries of Python, time information is added into packets to calculate end to end delay. "threading" and "socket" libraries of Python are used to implement TCP and UDP at the application level and to create multithreaded structure.

B. Motivation

This project was helpful for us to learn and experience topics like socket programming with different transport layer protocols, scripting languages and multithreaded structures. We have learned different terms and tools related with stream, packets, protocols and network delay. We have also practiced how to measure or manipulate them.

In addition, we have become familiar with the roles of different nodes in networks such as router, broker, receiver and sender nodes.

III. Experimental Results

A. Tools Used For Experiments

NTP: NTP (Network Time Protocol) is a protocol that is used to synchronize clock of computer systems overpacket-switched networks. We used NTP because sending the data computerclock is in nanoseconds, but in network systems packets are sent probably in microseconds, so the packets are received before its time because of the asynchronization.

Netem: NetEm is an enhancement of the Linux traffic control facilities that allow to add delay, packet loss, duplication and more other characteristics to packets outgoing from a selected network interface. In our experiments we used this command for emulating the delay loss and other similar properties of our network. To achieve this, we applied various kinds of this command for each network interface of the nodes and with some parameters for the virtual machines we created in GENI.

B. *Experimental Part*

For this part, we mainly made three experiments and plot the information that we gather from these experiments. Since we were needed to repeat the experiment for a considerable amount of time to get meaningful results, we repeated execution exactly 100 times for each experiment. We have achieved that with the help of the file that we created with a bash script on the implementation phase. That file was containing exactly 102400 bytes which is equal to $102400 = 1024 * 100$ bytes. With this kind of design, since in source node we were just reading data from file as 1024 bytes, execution was getting repeated 100 times.

Main task was to calculate end-to-end delays between source and destination nodes, and as mentioned earlier on this report, to achieve this, destination node was calculating each packet's end-to-end delay upon combining the chunks that construct it, then it was storing it to a container. Since we execute our sending data from source to destination process 100 times, in that container we get 100 end-to-end delays and after sending process terminates destination node calculates average end-to-end delay with taking the mean of the end-to-end delays in that container.

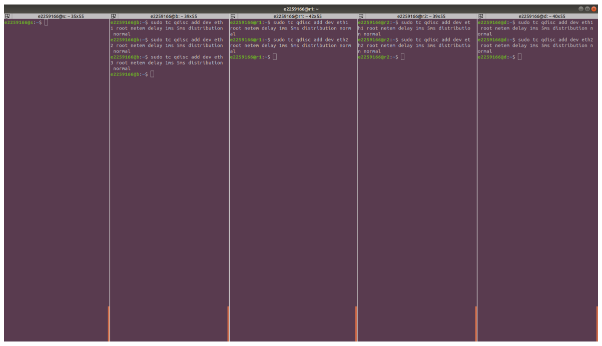
For the experiments, firstly we executed our process for 0 network emulation delay, then we executed it for $1 + -5$ normally distributed network emulation delay, afterwards we executed it for $20 + -5$ normally distributed network emulation delay and finally we executed it for $60 + -5$ normally distributed network emulation delay. The commands that we used for delete, show and add for each experiments are indicated on the README file.

On Figure 3 a screenshot of example usage of netem/tc command in our experiments can be seen. On Figure 4,5,6,7 screenshots of the results for the each experiment can be seen clearly (Average end-to-end delays are shown in the rightmost terminal's last line).

To plot graph of end-to-end delay vs network emulation delay with our gatherings from experiments, We also needed to calculate error domains with 95% Confidence Interval. To achieve this goal, we used following formula for each of the experiments:

$$e = z \times \frac{\text{(standard deviation for that experiment's delays)}}{\sqrt{\text{number of executions}}}$$

$$= 1.960 \times \frac{\text{(standard deviation for that experiment's delays)}}{\sqrt{100}}$$



```
e2259166@r1: ~
```

```
e2259166@s: ~ 35x55
e2259166@s:~$ 
```

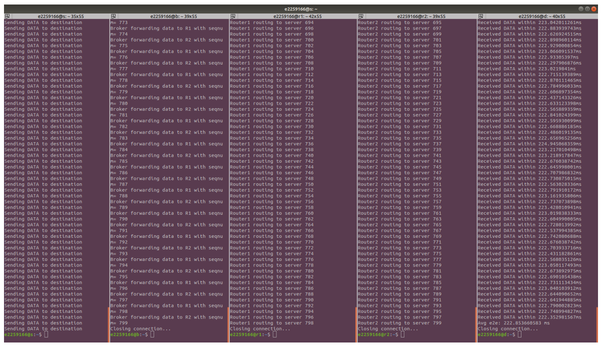
```
e2259166@b: ~ 39x55
e2259166@b:~$ sudo tc qdisc add dev eth
1 root netem delay 1ms 5ms distribution
 normal
e2259166@b:~$ sudo tc qdisc add dev eth
2 root netem delay 1ms 5ms distribution
 normal
e2259166@b:~$ sudo tc qdisc add dev eth
3 root netem delay 1ms 5ms distribution
 normal
e2259166@b:~$ 
```

```
e2259166@r1: ~ 42x55
e2259166@r1:~$ sudo tc qdisc add dev eth1
root netem delay 1ms 5ms distribution norm
al
e2259166@r1:~$ sudo tc qdisc add dev eth2
root netem delay 1ms 5ms distribution norm
al
e2259166@r1:~$ 
```

```
e2259166@r2: ~ 39x55
e2259166@r2:~$ sudo tc qdisc add dev et
h1 root netem delay 1ms 5ms distributio
n normal
e2259166@r2:~$ sudo tc qdisc add dev et
h2 root netem delay 1ms 5ms distributio
n normal
e2259166@r2:~$ 
```

```
e2259166@d: ~ 40x55
e2259166@d:~$ sudo tc qdisc add dev eth1
 root netem delay 1ms 5ms distribution n
ormal
e2259166@d:~$ sudo tc qdisc add dev eth2
 root netem delay 1ms 5ms distribution n
ormal
e2259166@d:~$ 
```

Fig. 3. Example usage of netem/tc command on one our experiments (Shown just for the first experiment)



```
e2259166@s: ~
```

```
e2259166@s: ~ 35x55
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
e2259166@s:~$ 
```

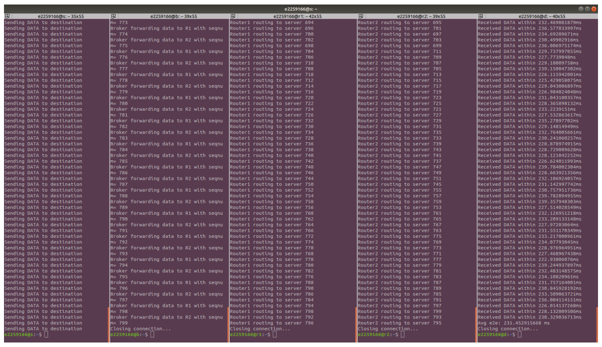
```
e2259166@b: ~ 39x55
m= 773
Broker forwarding data to R1 with seqnu
m= 774
Broker forwarding data to R2 with seqnu
m= 775
Broker forwarding data to R1 with seqnu
m= 776
Broker forwarding data to R2 with seqnu
m= 777
Broker forwarding data to R1 with seqnu
m= 778
Broker forwarding data to R2 with seqnu
m= 779
Broker forwarding data to R1 with seqnu
m= 780
Broker forwarding data to R2 with seqnu
m= 781
Broker forwarding data to R1 with seqnu
m= 782
Broker forwarding data to R2 with seqnu
m= 783
Broker forwarding data to R1 with seqnu
m= 784
Broker forwarding data to R2 with seqnu
m= 785
Broker forwarding data to R1 with seqnu
m= 786
Broker forwarding data to R2 with seqnu
m= 787
Broker forwarding data to R1 with seqnu
m= 788
Broker forwarding data to R2 with seqnu
m= 789
Broker forwarding data to R1 with seqnu
m= 790
Broker forwarding data to R2 with seqnu
m= 791
Broker forwarding data to R1 with seqnu
m= 792
Broker forwarding data to R2 with seqnu
m= 793
Broker forwarding data to R1 with seqnu
m= 794
Broker forwarding data to R2 with seqnu
m= 795
Broker forwarding data to R1 with seqnu
m= 796
Broker forwarding data to R2 with seqnu
m= 797
Broker forwarding data to R1 with seqnu
m= 798
Broker forwarding data to R2 with seqnu
m= 799
Closing connection...
e2259166@b:~$ 
```

```
e2259166@r1: ~ 42x55
Router1 routing to server 694
Router1 routing to server 696
Router1 routing to server 698
Router1 routing to server 700
Router1 routing to server 702
Router1 routing to server 704
Router1 routing to server 706
Router1 routing to server 708
Router1 routing to server 710
Router1 routing to server 712
Router1 routing to server 714
Router1 routing to server 716
Router1 routing to server 718
Router1 routing to server 720
Router1 routing to server 722
Router1 routing to server 724
Router1 routing to server 726
Router1 routing to server 728
Router1 routing to server 730
Router1 routing to server 732
Router1 routing to server 734
Router1 routing to server 736
Router1 routing to server 738
Router1 routing to server 740
Router1 routing to server 742
Router1 routing to server 744
Router1 routing to server 746
Router1 routing to server 748
Router1 routing to server 750
Router1 routing to server 752
Router1 routing to server 754
Router1 routing to server 756
Router1 routing to server 758
Router1 routing to server 760
Router1 routing to server 762
Router1 routing to server 764
Router1 routing to server 766
Router1 routing to server 768
Router1 routing to server 770
Router1 routing to server 772
Router1 routing to server 774
Router1 routing to server 776
Router1 routing to server 778
Router1 routing to server 780
Router1 routing to server 782
Router1 routing to server 784
Router1 routing to server 786
Router1 routing to server 788
Router1 routing to server 790
Router1 routing to server 792
Router1 routing to server 794
Router1 routing to server 796
Router1 routing to server 798
Closing connection...
e2259166@r1:~$ 
```

```
e2259166@r2: ~ 39x55
Router2 routing to server 695
Router2 routing to server 697
Router2 routing to server 699
Router2 routing to server 701
Router2 routing to server 703
Router2 routing to server 705
Router2 routing to server 707
Router2 routing to server 709
Router2 routing to server 711
Router2 routing to server 713
Router2 routing to server 715
Router2 routing to server 717
Router2 routing to server 719
Router2 routing to server 721
Router2 routing to server 723
Router2 routing to server 725
Router2 routing to server 727
Router2 routing to server 729
Router2 routing to server 731
Router2 routing to server 733
Router2 routing to server 735
Router2 routing to server 737
Router2 routing to server 739
Router2 routing to server 741
Router2 routing to server 743
Router2 routing to server 745
Router2 routing to server 747
Router2 routing to server 749
Router2 routing to server 751
Router2 routing to server 753
Router2 routing to server 755
Router2 routing to server 757
Router2 routing to server 759
Router2 routing to server 761
Router2 routing to server 763
Router2 routing to server 765
Router2 routing to server 767
Router2 routing to server 769
Router2 routing to server 771
Router2 routing to server 773
Router2 routing to server 775
Router2 routing to server 777
Router2 routing to server 779
Router2 routing to server 781
Router2 routing to server 783
Router2 routing to server 785
Router2 routing to server 787
Router2 routing to server 789
Router2 routing to server 791
Router2 routing to server 793
Router2 routing to server 795
Router2 routing to server 797
Router2 routing to server 799
Closing connection...
e2259166@r2:~$ 
```

```
e2259166@d: ~ 40x55
Received DATA within 223.042011261ms
Received DATA within 222.883939743ms
Received DATA within 222.626924515ms
Received DATA within 222.898960114ms
Received DATA within 222.929000854ms
Received DATA within 223.066091537ms
Received DATA within 222.93305397ms
Received DATA within 222.297906876ms
Received DATA within 223.0219841ms
Received DATA within 222.715139389ms
Received DATA within 222.870111465ms
Received DATA within 222.784996033ms
Received DATA within 222.606897354ms
Received DATA within 222.437143326ms
Received DATA within 222.633123398ms
Received DATA within 222.565889359ms
Received DATA within 222.841024399ms
Received DATA within 222.595930099ms
Received DATA within 222.668886185ms
Received DATA within 222.486019135ms
Received DATA within 222.656965256ms
Received DATA within 224.945068359ms
Received DATA within 223.217010498ms
Received DATA within 223.218917847ms
Received DATA within 222.676038742ms
Received DATA within 222.645998001ms
Received DATA within 222.707986832ms
Received DATA within 222.730875015ms
Received DATA within 222.563028336ms
Received DATA within 222.791910172ms
Received DATA within 223.161935806ms
Received DATA within 222.737073898ms
Received DATA within 223.428010941ms
Received DATA within 223.019838333ms
Received DATA within 222.604990005ms
Received DATA within 222.728013992ms
Received DATA within 222.537994385ms
Received DATA within 222.742080688ms
Received DATA within 222.676038742ms
Received DATA within 222.703933716ms
Received DATA within 222.431182861ms
Received DATA within 222.568035126ms
Received DATA within 223.050117493ms
Received DATA within 222.673892975ms
Received DATA within 222.690105438ms
Received DATA within 222.731113434ms
Received DATA within 223.040103912ms
Received DATA within 222.644090652ms
Received DATA within 222.641944885ms
Received DATA within 222.790002823ms
Received DATA within 222.748994827ms
Received DATA within 222.352981567ms
Avg e2e: 222.853660583 ms
Closing connection...
e2259166@d:~$ 
```

Fig. 4. Average end-to-end delay result with No network emulation delay (Shown in rightmost terminal's last line)



```
e2259166@s: ~
```

```
e2259166@s: ~ 35x55
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
e2259166@s:~$ 
```

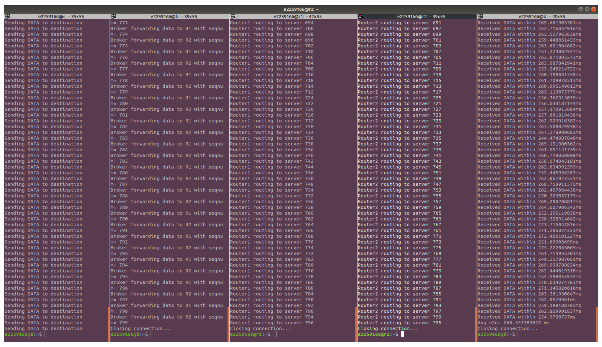
```
e2259166@b: ~ 39x55
m= 773
Broker forwarding data to R1 with seqnu
m= 774
Broker forwarding data to R2 with seqnu
m= 775
Broker forwarding data to R1 with seqnu
m= 776
Broker forwarding data to R2 with seqnu
m= 777
Broker forwarding data to R1 with seqnu
m= 778
Broker forwarding data to R2 with seqnu
m= 779
Broker forwarding data to R1 with seqnu
m= 780
Broker forwarding data to R2 with seqnu
m= 781
Broker forwarding data to R1 with seqnu
m= 782
Broker forwarding data to R2 with seqnu
m= 783
Broker forwarding data to R1 with seqnu
m= 784
Broker forwarding data to R2 with seqnu
m= 785
Broker forwarding data to R1 with seqnu
m= 786
Broker forwarding data to R2 with seqnu
m= 787
Broker forwarding data to R1 with seqnu
m= 788
Broker forwarding data to R2 with seqnu
m= 789
Broker forwarding data to R1 with seqnu
m= 790
Broker forwarding data to R2 with seqnu
m= 791
Broker forwarding data to R1 with seqnu
m= 792
Broker forwarding data to R2 with seqnu
m= 793
Broker forwarding data to R1 with seqnu
m= 794
Broker forwarding data to R2 with seqnu
m= 795
Broker forwarding data to R1 with seqnu
m= 796
Broker forwarding data to R2 with seqnu
m= 797
Broker forwarding data to R1 with seqnu
m= 798
Broker forwarding data to R2 with seqnu
m= 799
Closing connection...
e2259166@b:~$ 
```

```
e2259166@r1: ~ 42x55
Router1 routing to server 694
Router1 routing to server 696
Router1 routing to server 700
Router1 routing to server 702
Router1 routing to server 698
Router1 routing to server 704
Router1 routing to server 706
Router1 routing to server 710
Router1 routing to server 708
Router1 routing to server 718
Router1 routing to server 712
Router1 routing to server 714
Router1 routing to server 716
Router1 routing to server 720
Router1 routing to server 722
Router1 routing to server 724
Router1 routing to server 726
Router1 routing to server 732
Router1 routing to server 730
Router1 routing to server 734
Router1 routing to server 728
Router1 routing to server 736
Router1 routing to server 738
Router1 routing to server 740
Router1 routing to server 742
Router1 routing to server 748
Router1 routing to server 746
Router1 routing to server 744
Router1 routing to server 750
Router1 routing to server 752
Router1 routing to server 754
Router1 routing to server 758
Router1 routing to server 756
Router1 routing to server 760
Router1 routing to server 762
Router1 routing to server 764
Router1 routing to server 766
Router1 routing to server 772
Router1 routing to server 774
Router1 routing to server 770
Router1 routing to server 768
Router1 routing to server 778
Router1 routing to server 780
Router1 routing to server 782
Router1 routing to server 776
Router1 routing to server 788
Router1 routing to server 790
Router1 routing to server 786
Router1 routing to server 784
Router1 routing to server 794
Router1 routing to server 798
Router1 routing to server 792
Router1 routing to server 796
Closing connection...
e2259166@r1:~$ 
```

```
e2259166@r2: ~ 39x55
Router2 routing to server 695
Router2 routing to server 701
Router2 routing to server 697
Router2 routing to server 703
Router2 routing to server 699
Router2 routing to server 711
Router2 routing to server 709
Router2 routing to server 707
Router2 routing to server 705
Router2 routing to server 713
Router2 routing to server 715
Router2 routing to server 717
Router2 routing to server 719
Router2 routing to server 723
Router2 routing to server 725
Router2 routing to server 721
Router2 routing to server 727
Router2 routing to server 729
Router2 routing to server 731
Router2 routing to server 735
Router2 routing to server 733
Router2 routing to server 739
Router2 routing to server 743
Router2 routing to server 741
Router2 routing to server 737
Router2 routing to server 747
Router2 routing to server 749
Router2 routing to server 751
Router2 routing to server 745
Router2 routing to server 755
Router2 routing to server 757
Router2 routing to server 759
Router2 routing to server 753
Router2 routing to server 761
Router2 routing to server 765
Router2 routing to server 767
Router2 routing to server 763
Router2 routing to server 775
Router2 routing to server 769
Router2 routing to server 773
Router2 routing to server 771
Router2 routing to server 777
Router2 routing to server 779
Router2 routing to server 781
Router2 routing to server 783
Router2 routing to server 787
Router2 routing to server 789
Router2 routing to server 785
Router2 routing to server 791
Router2 routing to server 797
Router2 routing to server 799
Router2 routing to server 793
Router2 routing to server 795
Closing connection...
e2259166@r2:~$ 
```

```
e2259166@d: ~ 40x55
Received DATA within 232.469081879ms
Received DATA within 236.577033997ms
Received DATA within 234.69209671ms
Received DATA within 230.49902916ms
Received DATA within 236.006975174ms
Received DATA within 229.737997055ms
Received DATA within 227.7739048ms
Received DATA within 229.18009758ms
Received DATA within 239.150047302ms
Received DATA within 226.115942001ms
Received DATA within 225.429058075ms
Received DATA within 229.043006897ms
Received DATA within 226.984024048ms
Received DATA within 231.043100357ms
Received DATA within 228.365898132ms
Received DATA within 233.2239151ms
Received DATA within 227.532863617ms
Received DATA within 235.27097702ms
Received DATA within 229.14814949ms
Received DATA within 232.764005661ms
Received DATA within 230.241060257ms
Received DATA within 228.878974915ms
Received DATA within 228.729009628ms
Received DATA within 228.121042252ms
Received DATA within 226.624011993ms
Received DATA within 239.784002304ms
Received DATA within 228.663921356ms
Received DATA within 232.106924057ms
Received DATA within 231.142997742ms
Received DATA within 230.757951736ms
Received DATA within 239.782094955ms
Received DATA within 239.357948303ms
Received DATA within 227.514028549ms
Received DATA within 232.126951218ms
Received DATA within 233.209133148ms
Received DATA within 227.97203064ms
Received DATA within 231.551170349ms
Received DATA within 233.217000961ms
Received DATA within 234.07793045ms
Received DATA within 228.976964951ms
Received DATA within 227.468967438ms
Received DATA within 222.93806076ms
Received DATA within 239.244937897ms
Received DATA within 232.483148575ms
Received DATA within 234.10820961ms
Received DATA within 231.757164001ms
Received DATA within 230.845928192ms
Received DATA within 233.509063721ms
Received DATA within 236.004114151ms
Received DATA within 226.014137268ms
Received DATA within 228.132009506ms
Received DATA within 230.329036713ms
Avg e2e: 231.452915668 ms
Closing connection...
e2259166@d:~$ 
```

Fig. 5. Average end-to-end delay result with 1+-5 network emulation delay (Shown in rightmost terminal's last line)



```
e2259166@r2: ~
```

```
e2259166@s: ~ 35x55
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
e2259166@s:~$ 
```

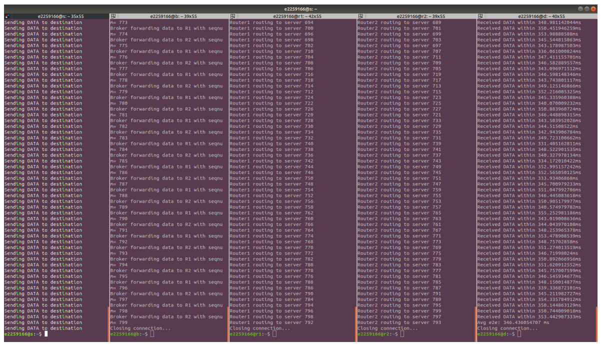
```
e2259166@b: ~ 39x55
m= 773
Broker forwarding data to R1 with seqnu
m= 774
Broker forwarding data to R2 with seqnu
m= 775
Broker forwarding data to R1 with seqnu
m= 776
Broker forwarding data to R2 with seqnu
m= 777
Broker forwarding data to R1 with seqnu
m= 778
Broker forwarding data to R2 with seqnu
m= 779
Broker forwarding data to R1 with seqnu
m= 780
Broker forwarding data to R2 with seqnu
m= 781
Broker forwarding data to R1 with seqnu
m= 782
Broker forwarding data to R2 with seqnu
m= 783
Broker forwarding data to R1 with seqnu
m= 784
Broker forwarding data to R2 with seqnu
m= 785
Broker forwarding data to R1 with seqnu
m= 786
Broker forwarding data to R2 with seqnu
m= 787
Broker forwarding data to R1 with seqnu
m= 788
Broker forwarding data to R2 with seqnu
m= 789
Broker forwarding data to R1 with seqnu
m= 790
Broker forwarding data to R2 with seqnu
m= 791
Broker forwarding data to R1 with seqnu
m= 792
Broker forwarding data to R2 with seqnu
m= 793
Broker forwarding data to R1 with seqnu
m= 794
Broker forwarding data to R2 with seqnu
m= 795
Broker forwarding data to R1 with seqnu
m= 796
Broker forwarding data to R2 with seqnu
m= 797
Broker forwarding data to R1 with seqnu
m= 798
Broker forwarding data to R2 with seqnu
m= 799
Closing connection...
e2259166@b:~$ 
```

```
e2259166@r1: ~ 42x55
Router1 routing to server 694
Router1 routing to server 700
Router1 routing to server 698
Router1 routing to server 696
Router1 routing to server 702
Router1 routing to server 710
Router1 routing to server 706
Router1 routing to server 704
Router1 routing to server 708
Router1 routing to server 716
Router1 routing to server 718
Router1 routing to server 714
Router1 routing to server 712
Router1 routing to server 724
Router1 routing to server 722
Router1 routing to server 720
Router1 routing to server 726
Router1 routing to server 732
Router1 routing to server 728
Router1 routing to server 734
Router1 routing to server 730
Router1 routing to server 738
Router1 routing to server 736
Router1 routing to server 740
Router1 routing to server 742
Router1 routing to server 744
Router1 routing to server 746
Router1 routing to server 750
Router1 routing to server 748
Router1 routing to server 754
Router1 routing to server 752
Router1 routing to server 756
Router1 routing to server 758
Router1 routing to server 760
Router1 routing to server 762
Router1 routing to server 764
Router1 routing to server 766
Router1 routing to server 768
Router1 routing to server 770
Router1 routing to server 774
Router1 routing to server 772
Router1 routing to server 782
Router1 routing to server 780
Router1 routing to server 776
Router1 routing to server 778
Router1 routing to server 786
Router1 routing to server 788
Router1 routing to server 784
Router1 routing to server 790
Router1 routing to server 792
Router1 routing to server 798
Router1 routing to server 794
Router1 routing to server 796
Closing connection...
e2259166@r1:~$ 
```

```
e2259166@r2: ~ 39x55
Router2 routing to server 691
Router2 routing to server 697
Router2 routing to server 699
Router2 routing to server 701
Router2 routing to server 703
Router2 routing to server 707
Router2 routing to server 705
Router2 routing to server 711
Router2 routing to server 709
Router2 routing to server 719
Router2 routing to server 715
Router2 routing to server 713
Router2 routing to server 717
Router2 routing to server 725
Router2 routing to server 721
Router2 routing to server 727
Router2 routing to server 723
Router2 routing to server 729
Router2 routing to server 731
Router2 routing to server 733
Router2 routing to server 735
Router2 routing to server 737
Router2 routing to server 739
Router2 routing to server 741
Router2 routing to server 743
Router2 routing to server 745
Router2 routing to server 751
Router2 routing to server 749
Router2 routing to server 747
Router2 routing to server 753
Router2 routing to server 755
Router2 routing to server 757
Router2 routing to server 759
Router2 routing to server 765
Router2 routing to server 763
Router2 routing to server 767
Router2 routing to server 761
Router2 routing to server 771
Router2 routing to server 773
Router2 routing to server 775
Router2 routing to server 769
Router2 routing to server 777
Router2 routing to server 781
Router2 routing to server 779
Router2 routing to server 783
Router2 routing to server 789
Router2 routing to server 787
Router2 routing to server 785
Router2 routing to server 791
Router2 routing to server 793
Router2 routing to server 797
Router2 routing to server 799
Router2 routing to server 795
Closing connection...
e2259166@r2:~$ 
```

```
e2259166@d: ~ 40x55
Received DATA within 269.661903381ms
Received DATA within 262.716054916ms
Received DATA within 265.527963638ms
Received DATA within 259.44805145ms
Received DATA within 263.601064682ms
Received DATA within 257.224082947ms
Received DATA within 263.971805573ms
Received DATA within 265.087842941ms
Received DATA within 253.248214722ms
Received DATA within 260.198831558ms
Received DATA within 261.709928513ms
Received DATA within 260.995149612ms
Received DATA within 265.119075775ms
Received DATA within 256.362915039ms
Received DATA within 256.033182144ms
Received DATA within 257.170915604ms
Received DATA within 257.661819458ms
Received DATA within 262.929916382ms
Received DATA within 267.508029938ms
Received DATA within 267.278909683ms
Received DATA within 244.474887848ms
Received DATA within 269.191980362ms
Received DATA within 261.551141739ms
Received DATA within 260.719060898ms
Received DATA within 258.074045181ms
Received DATA within 262.634992599ms
Received DATA within 252.445936203ms
Received DATA within 262.867927551ms
Received DATA within 268.719911575ms
Received DATA within 262.407064438ms
Received DATA within 250.323057175ms
Received DATA within 269.290208817ms
Received DATA within 264.607906342ms
Received DATA within 252.195119858ms
Received DATA within 258.328914642ms
Received DATA within 243.715047836ms
Received DATA within 271.296024323ms
Received DATA within 272.406101227ms
Received DATA within 251.60908699ms
Received DATA within 275.222063065ms
Received DATA within 261.714935303ms
Received DATA within 249.217987061ms
Received DATA within 269.998788834ms
Received DATA within 262.444019318ms
Received DATA within 254.598855972ms
Received DATA within 270.034074783ms
Received DATA within 271.145820618ms
Received DATA within 263.361930847ms
Received DATA within 262.9570961ms
Received DATA within 259.198188782ms
Received DATA within 262.809991837ms
Received DATA within 254.9700737ms
Avg e2e: 260.553383827 ms
Closing connection...
e2259166@d:~$ 
```

Fig. 6. Average end-to-end delay result with 20+-5 network emulation delay (Shown in rightmost terminal's last line)



```
e2259166@s: ~ 35x55
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
Sending DATA to destination
e2259166@s:~$
```

```
e2259166@b: ~ 39x55
m= 773
Broker forwarding data to R1 with seqnu
m= 774
Broker forwarding data to R2 with seqnu
m= 775
Broker forwarding data to R1 with seqnu
m= 776
Broker forwarding data to R2 with seqnu
m= 777
Broker forwarding data to R1 with seqnu
m= 778
Broker forwarding data to R2 with seqnu
m= 779
Broker forwarding data to R1 with seqnu
m= 780
Broker forwarding data to R2 with seqnu
m= 781
Broker forwarding data to R1 with seqnu
m= 782
Broker forwarding data to R2 with seqnu
m= 783
Broker forwarding data to R1 with seqnu
m= 784
Broker forwarding data to R2 with seqnu
m= 785
Broker forwarding data to R1 with seqnu
m= 786
Broker forwarding data to R2 with seqnu
m= 787
Broker forwarding data to R1 with seqnu
m= 788
Broker forwarding data to R2 with seqnu
m= 789
Broker forwarding data to R1 with seqnu
m= 790
Broker forwarding data to R2 with seqnu
m= 791
Broker forwarding data to R1 with seqnu
m= 792
Broker forwarding data to R2 with seqnu
m= 793
Broker forwarding data to R1 with seqnu
m= 794
Broker forwarding data to R2 with seqnu
m= 795
Broker forwarding data to R1 with seqnu
m= 796
Broker forwarding data to R2 with seqnu
m= 797
Broker forwarding data to R1 with seqnu
m= 798
Broker forwarding data to R2 with seqnu
m= 799
Closing connection...
e2259166@b:~$
```

```
e2259166@r1: ~ 42x55
Router1 routing to server 694
Router1 routing to server 700
Router1 routing to server 696
Router1 routing to server 698
Router1 routing to server 702
Router1 routing to server 710
Router1 routing to server 704
Router1 routing to server 708
Router1 routing to server 706
Router1 routing to server 716
Router1 routing to server 718
Router1 routing to server 714
Router1 routing to server 712
Router1 routing to server 724
Router1 routing to server 722
Router1 routing to server 726
Router1 routing to server 720
Router1 routing to server 728
Router1 routing to server 730
Router1 routing to server 734
Router1 routing to server 732
Router1 routing to server 740
Router1 routing to server 738
Router1 routing to server 736
Router1 routing to server 742
Router1 routing to server 744
Router1 routing to server 746
Router1 routing to server 750
Router1 routing to server 748
Router1 routing to server 754
Router1 routing to server 752
Router1 routing to server 756
Router1 routing to server 758
Router1 routing to server 762
Router1 routing to server 760
Router1 routing to server 766
Router1 routing to server 764
Router1 routing to server 774
Router1 routing to server 768
Router1 routing to server 770
Router1 routing to server 772
Router1 routing to server 782
Router1 routing to server 780
Router1 routing to server 778
Router1 routing to server 776
Router1 routing to server 788
Router1 routing to server 786
Router1 routing to server 790
Router1 routing to server 784
Router1 routing to server 794
Router1 routing to server 796
Router1 routing to server 798
Router1 routing to server 792
Closing connection...
e2259166@r1:~$
```

```
e2259166@r2: ~ 39x55
Router2 routing to server 689
Router2 routing to server 701
Router2 routing to server 699
Router2 routing to server 703
Router2 routing to server 697
Router2 routing to server 707
Router2 routing to server 711
Router2 routing to server 709
Router2 routing to server 705
Router2 routing to server 719
Router2 routing to server 717
Router2 routing to server 713
Router2 routing to server 715
Router2 routing to server 723
Router2 routing to server 725
Router2 routing to server 727
Router2 routing to server 721
Router2 routing to server 733
Router2 routing to server 729
Router2 routing to server 735
Router2 routing to server 731
Router2 routing to server 739
Router2 routing to server 741
Router2 routing to server 737
Router2 routing to server 743
Router2 routing to server 749
Router2 routing to server 745
Router2 routing to server 751
Router2 routing to server 747
Router2 routing to server 759
Router2 routing to server 755
Router2 routing to server 753
Router2 routing to server 757
Router2 routing to server 765
Router2 routing to server 763
Router2 routing to server 761
Router2 routing to server 767
Router2 routing to server 771
Router2 routing to server 773
Router2 routing to server 769
Router2 routing to server 775
Router2 routing to server 779
Router2 routing to server 783
Router2 routing to server 777
Router2 routing to server 781
Router2 routing to server 785
Router2 routing to server 787
Router2 routing to server 791
Router2 routing to server 789
Router2 routing to server 795
Router2 routing to server 799
Router2 routing to server 797
Router2 routing to server 793
Closing connection...
e2259166@r2:~$
```

```
e2259166@d: ~ 40x55
Received DATA within 348.981142044ms
Received DATA within 350.451946259ms
Received DATA within 353.98888588ms
Received DATA within 345.544815063ms
Received DATA within 343.178987503ms
Received DATA within 336.061000824ms
Received DATA within 347.411155701ms
Received DATA within 346.582889557ms
Received DATA within 343.096971512ms
Received DATA within 346.598148346ms
Received DATA within 343.743801117ms
Received DATA within 349.125146866ms
Received DATA within 352.216005325ms
Received DATA within 345.335960388ms
Received DATA within 340.070009232ms
Received DATA within 350.883960724ms
Received DATA within 346.448898315ms
Received DATA within 343.503952026ms
Received DATA within 344.51508522ms
Received DATA within 342.943906784ms
Received DATA within 349.723100662ms
Received DATA within 331.405162811ms
Received DATA within 348.522901535ms
Received DATA within 340.327978134ms
Received DATA within 334.172010422ms
Received DATA within 352.995157242ms
Received DATA within 352.565050125ms
Received DATA within 333.93406868ms
Received DATA within 345.700979233ms
Received DATA within 351.047992706ms
Received DATA within 346.445083618ms
Received DATA within 350.905179977ms
Received DATA within 340.574979782ms
Received DATA within 355.252981186ms
Received DATA within 343.019008636ms
Received DATA within 344.043970108ms
Received DATA within 348.253965378ms
Received DATA within 353.478908539ms
Received DATA within 348.75702858ms
Received DATA within 351.274013519ms
Received DATA within 346.71998024ms
Received DATA within 350.892066956ms
Received DATA within 351.620912552ms
Received DATA within 345.757007599ms
Received DATA within 346.545934677ms
Received DATA within 348.150014877ms
Received DATA within 339.336872101ms
Received DATA within 345.211982727ms
Received DATA within 354.335784912ms
Received DATA within 350.144863129ms
Received DATA within 350.744009018ms
Received DATA within 353.442907333ms
Avg e2e: 346.436054707 ms
Closing connection...
e2259166@d:~$
```

Fig. 7. Average end-to-end delay result with 60+-5 network emulation delay (Shown in rightmost terminal's last line)

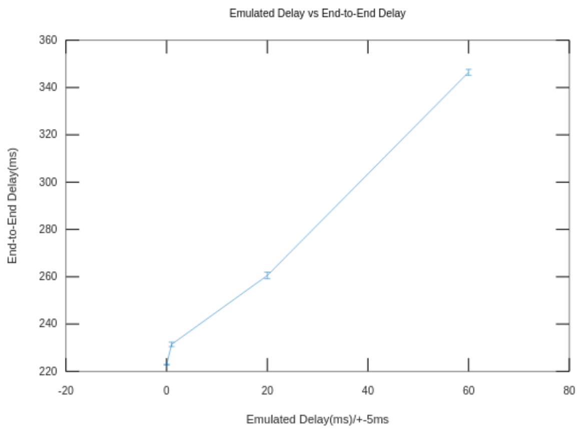


Fig. 8. Graph of end-to-end delay vs network emulation delay

Where e is error for that experiment, z is the confidence interval variable (for our case, confidence interval = 95% , so z = 1.960).

Upon calculating these errors, we plotted the end-to-end delay vs network emulation delay graph with the help of Octave. Figure 8 shows this graph clearly.

For 0 ms network emulation delay (normal execution of process), we get a result of 222 ms and for 1 ms network emulation delay we got a result of 231 ms. For 20 ms we get a result of 260 ms and for 60 ms network emulation delay we got a result of 346 ms. First thing we observe is that since we apply network emulation delay for only two links (broker to routers and routers to destination), the delay approximately doubles in the result for the original added delay. The second thing that we see is that resulting graph is a linear fashion increasing graph, namely, as network emulation delay increases end-to-end delay also increases.