FIles Submitted:
There was a small inconsistency in the naming convention(myfirst2.cc was used twice in the assignment doc). So i have named the files with #task.

1.**first_task1.cc** ->corresponding to 1st task
To compile and run : ns-3-dev$ ./waf --run 'first_task1.cc'
For 3rd task : ns-3-dev$ ./waf --run 'first_task1.cc --RngRun=3'

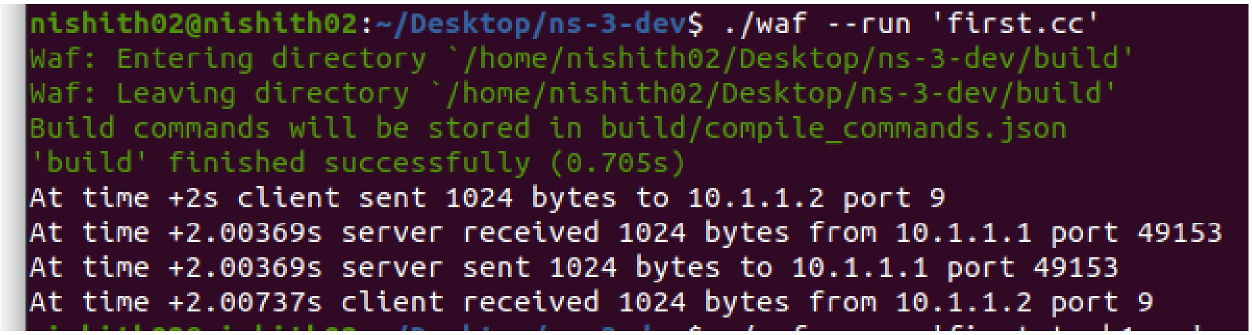
2.**first_task2.cc** ->corresponding to 2nd task
To compile and run : ns-3-dev$ ./waf --run 'first_task2.cc'

3.**first_task4.cc** ->corresponding to 4th task
To compile and run : ns-3-dev$ ./waf --run 'first_task4.cc --datarate="10Mbps" --delay="10ms"'

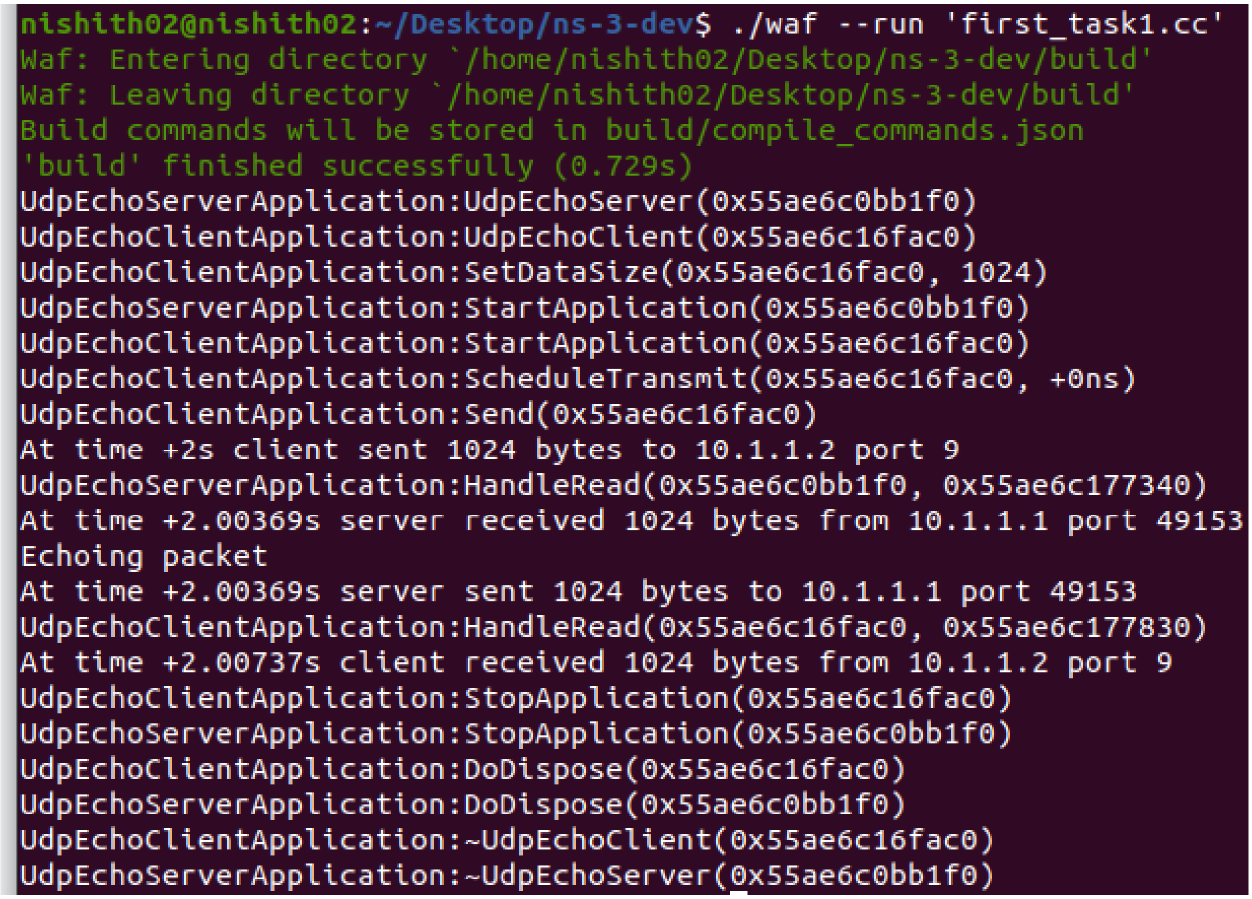
4.**first_task4b.cc** -> corresponding to 4b task(taking max packets from command)
To compile and run: ns-3-dev./waf --run 'first_task4b.cc --datarate="5Mbps" --delay="2ms" --maxpacket=3'

5.**first_task5.cc** ->corresponding to 5th task
To compile and run : ns-3-dev$ ./waf --run 'first_task5.cc --interval=1.0 --packetsize=1024'

Task 1 :



```
nishith02@nishith02:~/Desktop/ns-3-dev$ ./waf --run 'first.cc'
Waf: Entering directory `/home/nishith02/Desktop/ns-3-dev/build'
Waf: Leaving directory `/home/nishith02/Desktop/ns-3-dev/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.705s)
At time +2s client sent 1024 bytes to 10.1.1.2 port 9
At time +2.00369s server received 1024 bytes from 10.1.1.1 port 49153
At time +2.00369s server sent 1024 bytes to 10.1.1.1 port 49153
At time +2.00737s client received 1024 bytes from 10.1.1.2 port 9
```

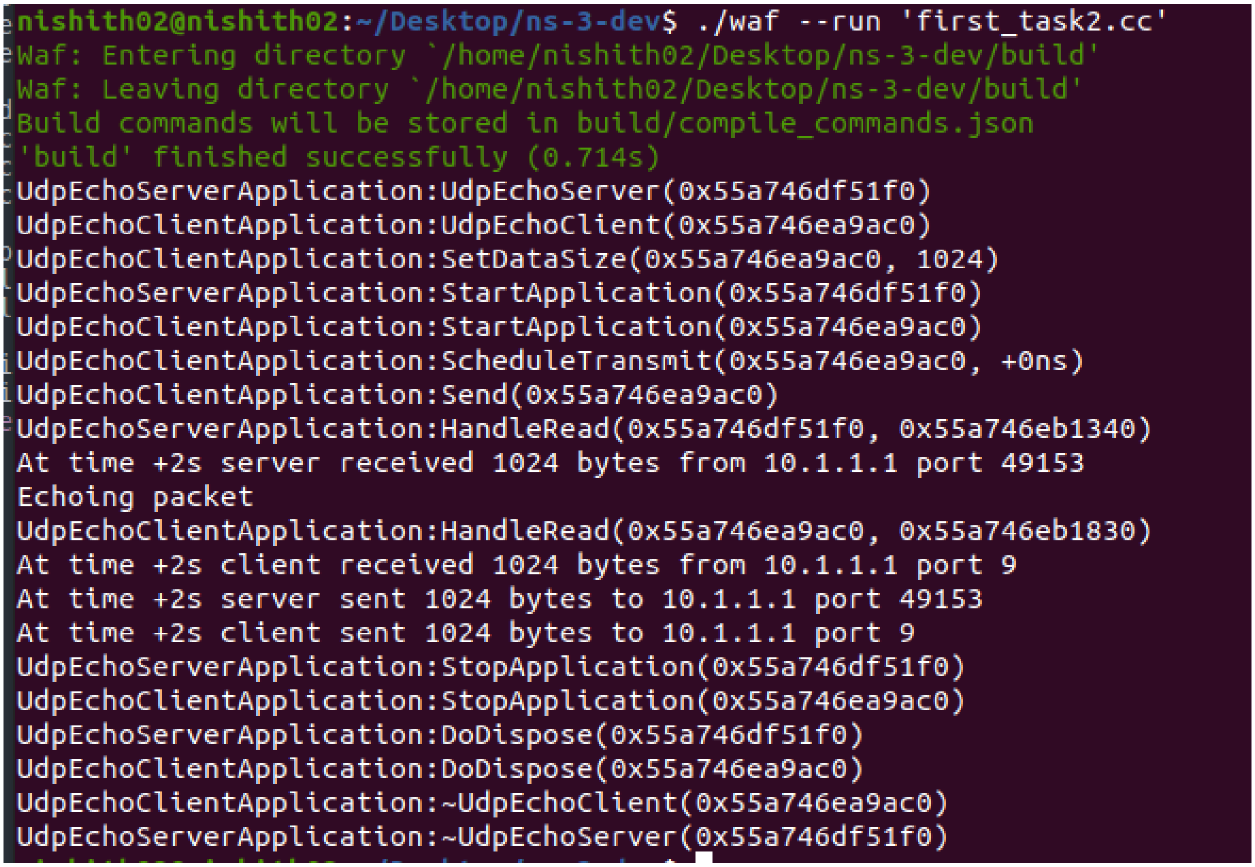


```
nishith02@nishith02:~/Desktop/ns-3-dev$ ./waf --run 'first_task1.cc'
Waf: Entering directory `/home/nishith02/Desktop/ns-3-dev/build'
Waf: Leaving directory `/home/nishith02/Desktop/ns-3-dev/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.729s)
UdpEchoServerApplication:UdpEchoServer(0x55ae6c0bb1f0)
UdpEchoClientApplication:UdpEchoClient(0x55ae6c16fac0)
UdpEchoClientApplication:SetDataSize(0x55ae6c16fac0, 1024)
UdpEchoServerApplication:StartApplication(0x55ae6c0bb1f0)
UdpEchoClientApplication:StartApplication(0x55ae6c16fac0)
UdpEchoClientApplication:ScheduleTransmit(0x55ae6c16fac0, +0ns)
UdpEchoClientApplication:Send(0x55ae6c16fac0)
At time +2s client sent 1024 bytes to 10.1.1.2 port 9
UdpEchoServerApplication:HandleRead(0x55ae6c0bb1f0, 0x55ae6c177340)
At time +2.00369s server received 1024 bytes from 10.1.1.1 port 49153
Echoing packet
At time +2.00369s server sent 1024 bytes to 10.1.1.1 port 49153
UdpEchoClientApplication:HandleRead(0x55ae6c16fac0, 0x55ae6c177830)
At time +2.00737s client received 1024 bytes from 10.1.1.2 port 9
UdpEchoClientApplication:StopApplication(0x55ae6c16fac0)
UdpEchoServerApplication:StopApplication(0x55ae6c0bb1f0)
UdpEchoClientApplication:DoDispose(0x55ae6c16fac0)
UdpEchoServerApplication:DoDispose(0x55ae6c0bb1f0)
UdpEchoClientApplication:~UdpEchoClient(0x55ae6c16fac0)
UdpEchoServerApplication:~UdpEchoServer(0x55ae6c0bb1f0)
```

The first photo belongs to “ **LOG_LEVEL_INFO**” and the second photo belongs to “ **LOG_LEVEL_ALL**”. Individual messages belong to a single severity class. Inorder to see messages at a given severity class and higher, logging levels are used. Using LOG_LEVEL_INFO messages of severity LOG_INFO(Informational) and above are produced whereas using LOG_LEVEL_ALL prints all messages of any severity. So because of that, in the first picture, only information regarding client and server was printed but in the second case everything that has happened has been logged to us.

Task 2:

Output:

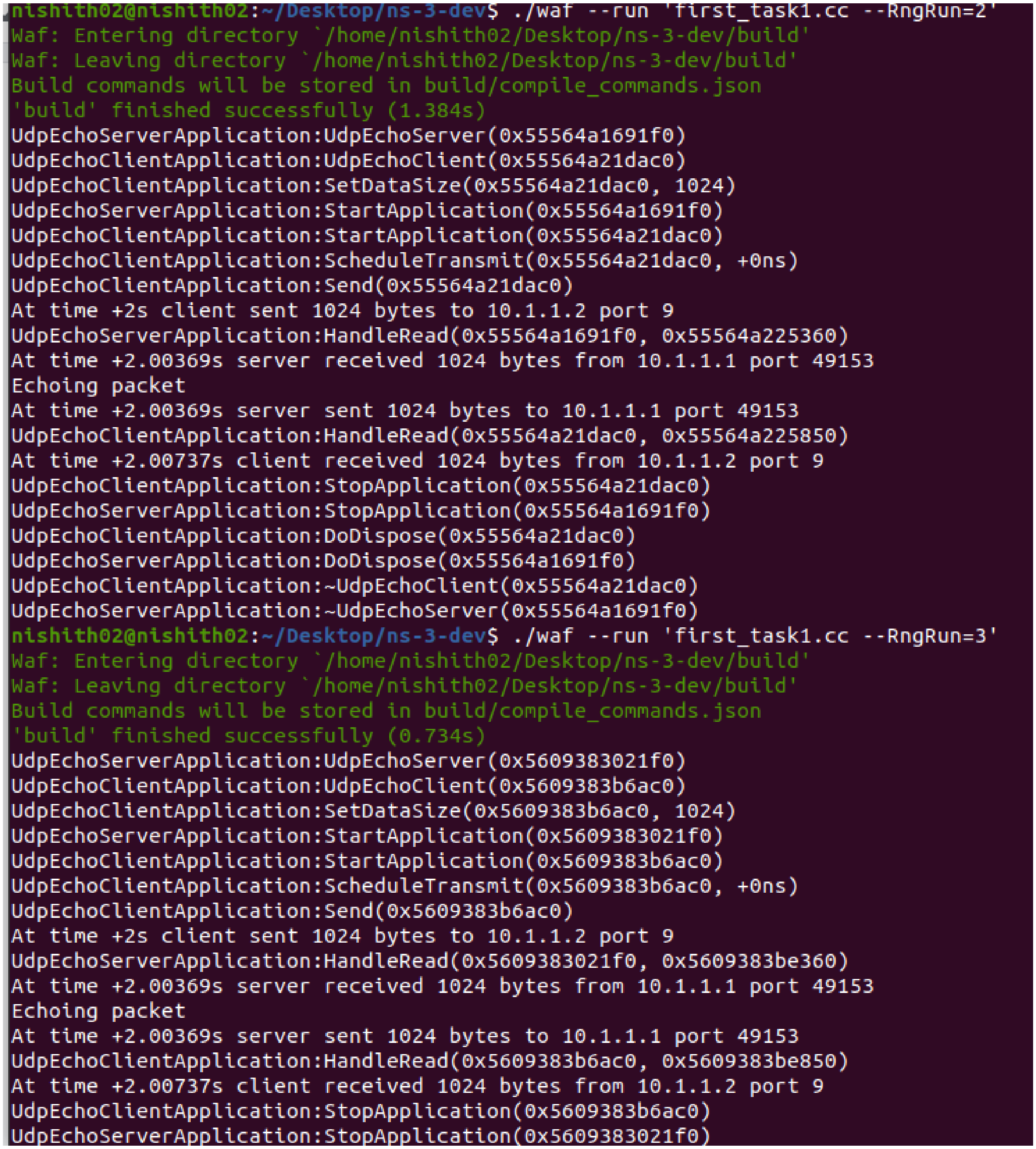


```
nishith02@nishith02:~/Desktop/ns-3-dev$ ./waf --run 'first_task2.cc'
Waf: Entering directory `/home/nishith02/Desktop/ns-3-dev/build'
Waf: Leaving directory `/home/nishith02/Desktop/ns-3-dev/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.714s)
UdpEchoServerApplication:UdpEchoServer(0x55a746df51f0)
UdpEchoClientApplication:UdpEchoClient(0x55a746ea9ac0)
UdpEchoClientApplication:SetDataSize(0x55a746ea9ac0, 1024)
UdpEchoServerApplication:StartApplication(0x55a746df51f0)
UdpEchoClientApplication:StartApplication(0x55a746ea9ac0)
UdpEchoClientApplication:ScheduleTransmit(0x55a746ea9ac0, +0ns)
UdpEchoClientApplication:Send(0x55a746ea9ac0)
UdpEchoServerApplication:HandleRead(0x55a746df51f0, 0x55a746eb1340)
At time +2s server received 1024 bytes from 10.1.1.1 port 49153
Echoing packet
UdpEchoClientApplication:HandleRead(0x55a746ea9ac0, 0x55a746eb1830)
At time +2s client received 1024 bytes from 10.1.1.1 port 9
At time +2s server sent 1024 bytes to 10.1.1.1 port 49153
At time +2s client sent 1024 bytes to 10.1.1.1 port 9
UdpEchoServerApplication:StopApplication(0x55a746df51f0)
UdpEchoClientApplication:StopApplication(0x55a746ea9ac0)
UdpEchoServerApplication:DoDispose(0x55a746df51f0)
UdpEchoClientApplication:DoDispose(0x55a746ea9ac0)
UdpEchoClientApplication:~UdpEchoClient(0x55a746ea9ac0)
UdpEchoServerApplication:~UdpEchoServer(0x55a746df51f0)
```

As it can be seen, there is no proper ordering of messages printed in the above output.(received is printed first, then the sent messages). Both sending and receiving has happened at t = 2 secs , which is expected as both the sender and receiver are the same.

Task 3:

There is no change in output when the seed is changed. This could be because there are no random numbers which are being used in myfirst.cc
(Picture of output below)



```
nishith02@nishith02:~/Desktop/ns-3-dev$ ./waf --run 'first_task1.cc --RngRun=2'
Waf: Entering directory `/home/nishith02/Desktop/ns-3-dev/build'
Waf: Leaving directory `/home/nishith02/Desktop/ns-3-dev/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (1.384s)
UdpEchoServerApplication:UdpEchoServer(0x55564a1691f0)
UdpEchoClientApplication:UdpEchoClient(0x55564a21dac0)
UdpEchoClientApplication:SetDataSize(0x55564a21dac0, 1024)
UdpEchoServerApplication:StartApplication(0x55564a1691f0)
UdpEchoClientApplication:StartApplication(0x55564a21dac0)
UdpEchoClientApplication:ScheduleTransmit(0x55564a21dac0, +0ns)
UdpEchoClientApplication:Send(0x55564a21dac0)
At time +2s client sent 1024 bytes to 10.1.1.2 port 9
UdpEchoServerApplication:HandleRead(0x55564a1691f0, 0x55564a225360)
At time +2.00369s server received 1024 bytes from 10.1.1.1 port 49153
Echoing packet
At time +2.00369s server sent 1024 bytes to 10.1.1.1 port 49153
UdpEchoClientApplication:HandleRead(0x55564a21dac0, 0x55564a225850)
At time +2.00737s client received 1024 bytes from 10.1.1.2 port 9
UdpEchoClientApplication:StopApplication(0x55564a21dac0)
UdpEchoServerApplication:StopApplication(0x55564a1691f0)
UdpEchoClientApplication:DoDispose(0x55564a21dac0)
UdpEchoServerApplication:DoDispose(0x55564a1691f0)
UdpEchoClientApplication:~UdpEchoClient(0x55564a21dac0)
UdpEchoServerApplication:~UdpEchoServer(0x55564a1691f0)
nishith02@nishith02:~/Desktop/ns-3-dev$ ./waf --run 'first_task1.cc --RngRun=3'
Waf: Entering directory `/home/nishith02/Desktop/ns-3-dev/build'
Waf: Leaving directory `/home/nishith02/Desktop/ns-3-dev/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.734s)
UdpEchoServerApplication:UdpEchoServer(0x5609383021f0)
UdpEchoClientApplication:UdpEchoClient(0x5609383b6ac0)
UdpEchoClientApplication:SetDataSize(0x5609383b6ac0, 1024)
UdpEchoServerApplication:StartApplication(0x5609383021f0)
UdpEchoClientApplication:StartApplication(0x5609383b6ac0)
UdpEchoClientApplication:ScheduleTransmit(0x5609383b6ac0, +0ns)
UdpEchoClientApplication:Send(0x5609383b6ac0)
At time +2s client sent 1024 bytes to 10.1.1.2 port 9
UdpEchoServerApplication:HandleRead(0x5609383021f0, 0x5609383be360)
At time +2.00369s server received 1024 bytes from 10.1.1.1 port 49153
Echoing packet
At time +2.00369s server sent 1024 bytes to 10.1.1.1 port 49153
UdpEchoClientApplication:HandleRead(0x5609383b6ac0, 0x5609383be850)
At time +2.00737s client received 1024 bytes from 10.1.1.2 port 9
UdpEchoClientApplication:StopApplication(0x5609383b6ac0)
UdpEchoServerApplication:StopApplication(0x5609383021f0)
```

Task 4:

Exp.No	DataRate(Mbps)	Delay(ms)	RTT(ms)
1	5	2	7.37
2	10	2	5.69
3	5	5	13.37
4	5	10	23.37
5	1	2	20.86
6	1	10	36.86
7	10	1	3.69
8	10	0.1	1.89
9	10	0	1.69
10	5	0	3.37

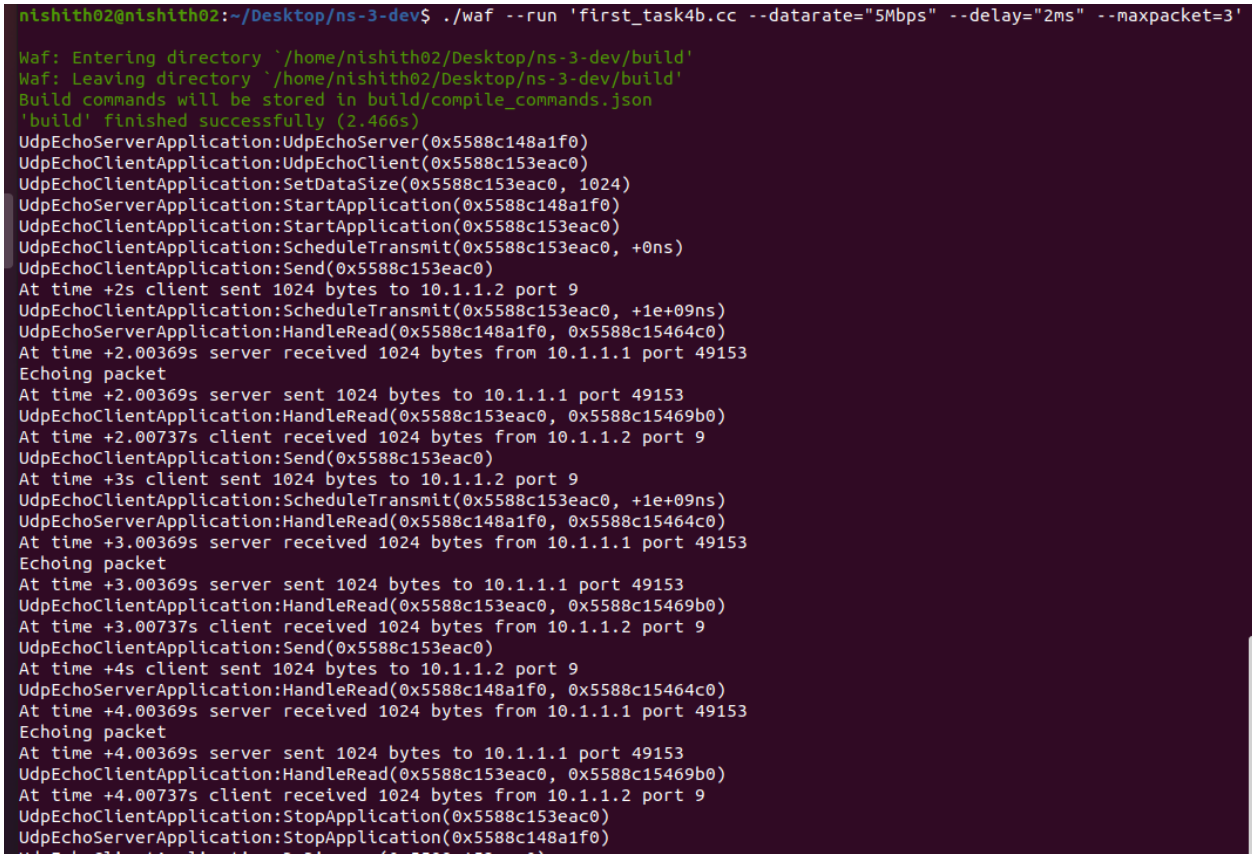
RTT = 2*transmission_time + 2*Propogation_delay (2x because echoing).
RTT increases whenever Data rate decreases or delay increases.

Consider observation 9, where data rate is 10 Mbps and delay is 0. This means RTT is 2 times the transmission time. Comparing this with observation 10, where data rate is 5 Mbps (half of that of observation 9), RTT becomes double as expected(as delay is 0 and speed is doubled).

Now compare observation 8 and 9, in both the cases data rate is the same. There is a delay of 0.1 ms. So RTT should increase by 2*0.1 secs, which is observed.

In all the above 10 observations, the expression for RTT holds.This seems like an ideal case point to point communication link, without any interference from other nodes from the network. RTT increases whenever Data rate decreases or delay increases.

4b)Client sends more than one packet :
When the client sends more than one packet, RTT does not change from packet to packet. Observe the below output, where data rate of 5Mbps, delay of 2 ms and 3 packets are given as input from command line. The RTT for all the 3 packets is 7.37 ms and which is the same which was obtained in the case of one packet.



```
nishith02@nishith02:~/Desktop/ns-3-dev$ ./waf --run 'first_task4b.cc --datarate="5Mbps" --delay="2ms" --maxpacket=3'

Waf: Entering directory `/home/nishith02/Desktop/ns-3-dev/build'
Waf: Leaving directory `/home/nishith02/Desktop/ns-3-dev/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (2.466s)
UdpEchoServerApplication:UdpEchoServer(0x5588c148a1f0)
UdpEchoClientApplication:UdpEchoClient(0x5588c153eac0)
UdpEchoClientApplication:SetDataSize(0x5588c153eac0, 1024)
UdpEchoServerApplication:StartApplication(0x5588c148a1f0)
UdpEchoClientApplication:StartApplication(0x5588c153eac0)
UdpEchoClientApplication:ScheduleTransmit(0x5588c153eac0, +0ns)
UdpEchoClientApplication:Send(0x5588c153eac0)
At time +2s client sent 1024 bytes to 10.1.1.2 port 9
UdpEchoClientApplication:ScheduleTransmit(0x5588c153eac0, +1e+09ns)
UdpEchoServerApplication:HandleRead(0x5588c148a1f0, 0x5588c15464c0)
At time +2.00369s server received 1024 bytes from 10.1.1.1 port 49153
Echoing packet
At time +2.00369s server sent 1024 bytes to 10.1.1.1 port 49153
UdpEchoClientApplication:HandleRead(0x5588c153eac0, 0x5588c15469b0)
At time +2.00737s client received 1024 bytes from 10.1.1.2 port 9
UdpEchoClientApplication:Send(0x5588c153eac0)
At time +3s client sent 1024 bytes to 10.1.1.2 port 9
UdpEchoClientApplication:ScheduleTransmit(0x5588c153eac0, +1e+09ns)
UdpEchoServerApplication:HandleRead(0x5588c148a1f0, 0x5588c15464c0)
At time +3.00369s server received 1024 bytes from 10.1.1.1 port 49153
Echoing packet
At time +3.00369s server sent 1024 bytes to 10.1.1.1 port 49153
UdpEchoClientApplication:HandleRead(0x5588c153eac0, 0x5588c15469b0)
At time +3.00737s client received 1024 bytes from 10.1.1.2 port 9
UdpEchoClientApplication:Send(0x5588c153eac0)
At time +4s client sent 1024 bytes to 10.1.1.2 port 9
UdpEchoServerApplication:HandleRead(0x5588c148a1f0, 0x5588c15464c0)
At time +4.00369s server received 1024 bytes from 10.1.1.1 port 49153
Echoing packet
At time +4.00369s server sent 1024 bytes to 10.1.1.1 port 49153
UdpEchoClientApplication:HandleRead(0x5588c153eac0, 0x5588c15469b0)
At time +4.00737s client received 1024 bytes from 10.1.1.2 port 9
UdpEchoClientApplication:StopApplication(0x5588c153eac0)
UdpEchoServerApplication:StopApplication(0x5588c148a1f0)
```

Task 5 :

Assuming, dataRate = 5Mbps, delay = 2 ms.

UDP echo sending rate = (2*packet_size)/interval (2 packets are being exchanged (echo) in a interval)

Exp.No	Interval(s)	PacketSize	UDP Echo Traffic Sending Rate (bps)	UDP Packet Loss Rate	RTT(ms)
1	1	1024	16384	0	7.37
2	2	1024	8192	0	7.37
3	1	2048	32768	0	10.72
4	2	2048	16384	0	10.72
5	2.5	4096	26214.4	0	17.34

6	2	512	4096	0	5.73

No packets were lost in the above experiments. As **packet size** increases , **RTT** increases as expected(keeping constant interval) but **interval time** is not having effect on the **RTT**(assuming constant packet size).

PLAGIARISM STATEMENT

I certify that this assignment/report is my own work, based on my personal study and/or research and that I have acknowledged all material and sources used in its preparation, whether they be books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. I also certify that this assignment/report has not previously been submitted for assessment in any other course, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that I have not copied in part or whole or otherwise plagiarised the work of other students and/or persons. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, I understand my responsibility to report honour violations by other students if I become aware of it.

Name: Sai Nishith Jupally

Date: 3-10-2021

Signature: JSN