

מלה 3 רשות ותקשרות

מגישים:

רונן יעקבוביץ
יובל מוסרי

ת.ז.:

212767214
213358039

Receiver.py:

First we establish a TCP connection between the sender and the receiver. Once the connection is established, the receiver waits for the sender to send the first part of the file. When the first part of the file is received, the receiver measures the time it took to receive it and saves this time.

The receiver then sends back an authentication to the sender, which includes changing the CC (Congestion Control) Algorithm used. After this, the receiver waits for the second part of the file to be received. Again, the receiver measures the time it took to receive the second part and saves this time.

If the receiver receives an exit message from the sender, the receiver prints out all the times that were saved during the process and calculates the average time for each part of the file received. The receiver then prints out the average time and closes the TCP connection.

If the receiver does not receive an exit message from the sender, the process goes back and waits for the next part of the file to be received.

(running example)

```
ron@ron-VirtualBox:~/Documents/Network3-main$ python3 Receiver.py
Got a connection from ('127.0.0.1', 57384)
First file has been received
Authentication has been sent.
Changing CC algorithem to reno.
Second file has been received
First file has been received
Authentication has been sent.
Changing CC algorithem to reno.
Second file has been received
First file has been received
Authentication has been sent.
Changing CC algorithem to reno.
Second file has been received
First file has been received
Authentication has been sent.
Changing CC algorithem to reno.
Second file has been received
#####
#####RESULTS#####
File-number      Cubic      Reno
seq = 0 cubic = 0.04778027534484863 rno = 0.058564186096191406
seq = 1 cubic = 0.004099130630493164 rno = 0.054575443267822266
seq = 2 cubic = 0.0041522979736328125 rno = 0.045096635818481445
seq = 3 cubic = 0.002798795700073242 rno = 0.0447688102722168
The avarge time for cubic is 0.014707624912261963
The avarge time for rno is 0.05075126886367798
ron@ron-VirtualBox:~/Documents/Network3-main$
```

Sender.py:

First the sender needs to read the file that we have created. Then the sender creates a TCP connection with the receiver. The sender then sends the first part of the file to the receiver.

At this point, the receiver checks for authentication. If the authentication is successful, the receiver changes the CC (Congestion Control) Algorithm used for the connection. The receiver then waits for the second part of the file to be sent by the sender.

After the second part of the file is sent by the sender, the user is given a decision. If the user decides to send the file again (for data gathering purposes), the sender notifies the receiver and changes the CC Algorithm back to the original one. The process then goes back, where the sender sends the first part of the file again.

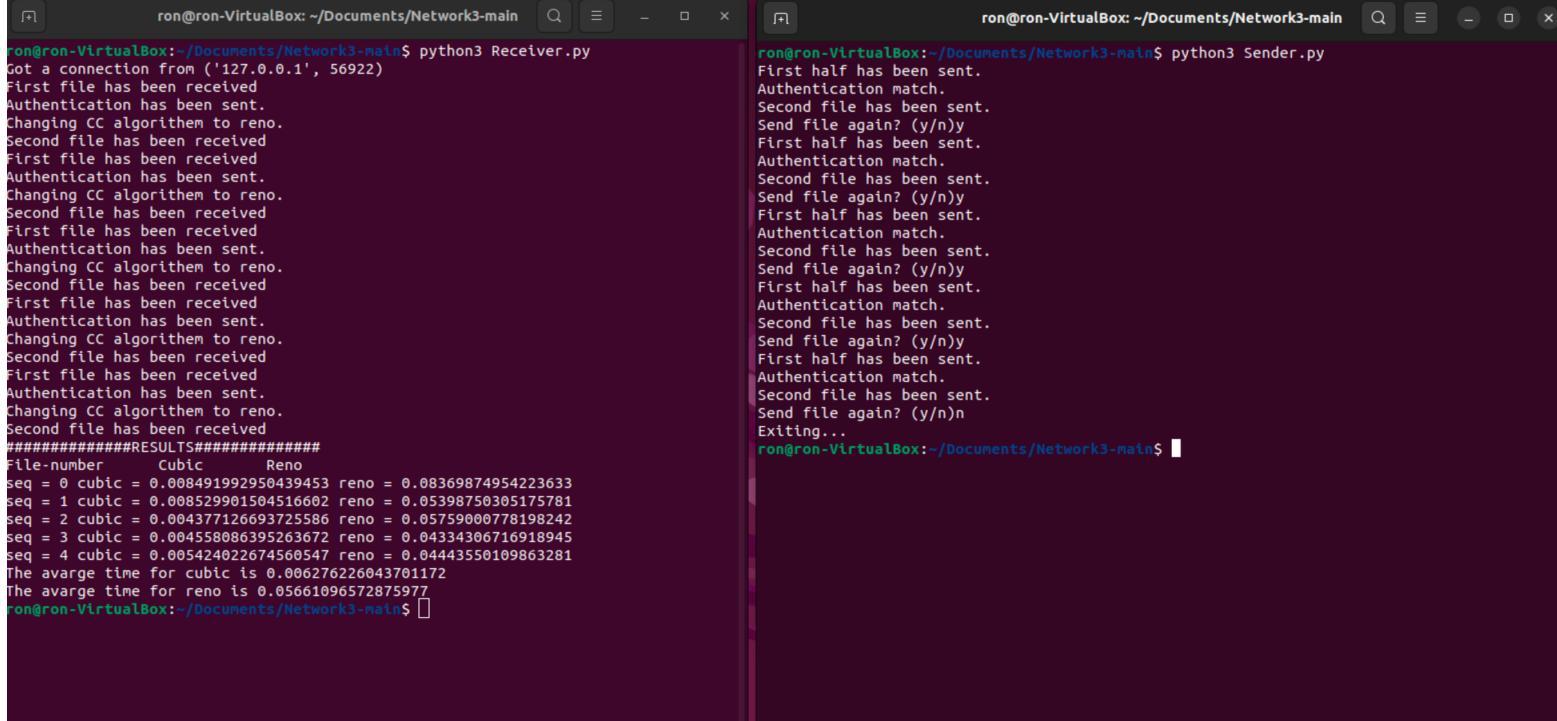
If the user decides not to send the file again, the sender sends an exit message to the receiver and closes the TCP connection.

(running example)

```
ron@ron-VirtualBox:~/Documents/Network3-main$ python3 Sender.py
First half has been sent.
Authentication match.
Second file has been sent.
Send file again? (y/n)y
First half has been sent.
Authentication match.
Second file has been sent.
Send file again? (y/n)y
First half has been sent.
Authentication match.
Second file has been sent.
Send file again? (y/n)y
First half has been sent.
Authentication match.
Second file has been sent.
Send file again? (y/n)n
Exiting...
ron@ron-VirtualBox:~/Documents/Network3-main$
```

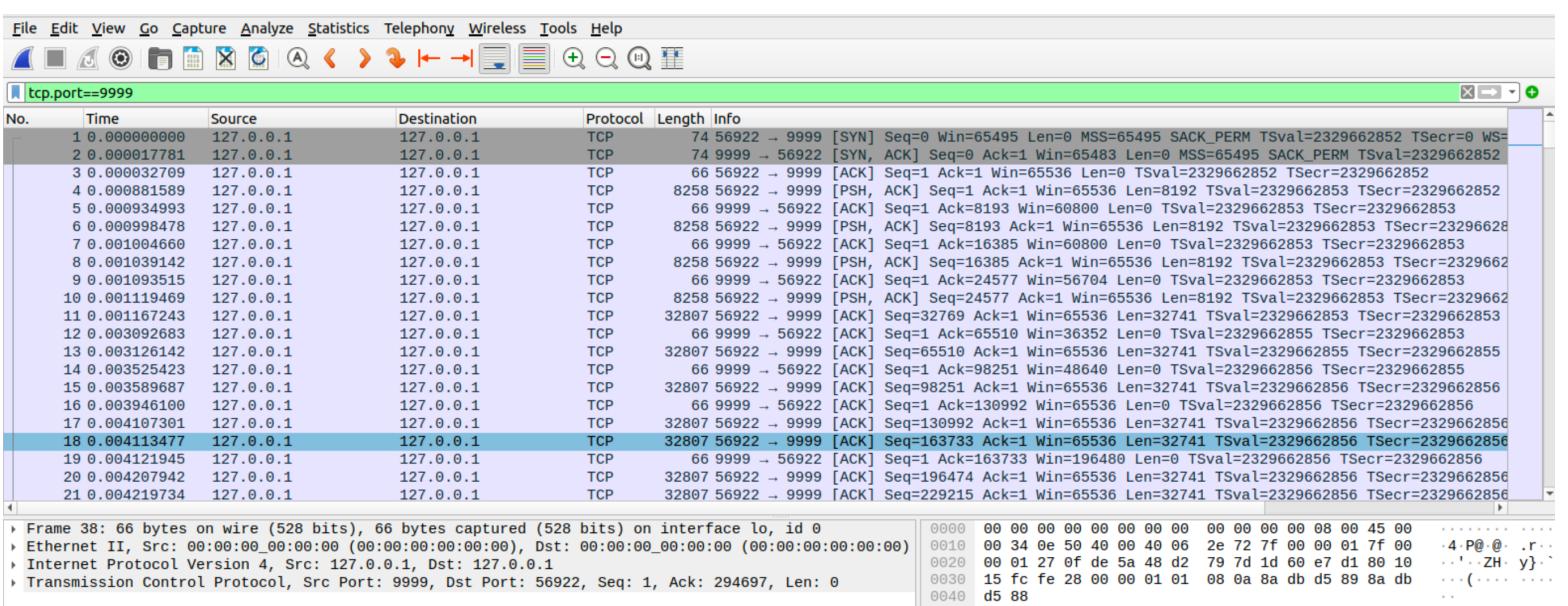
Packet loss percentage:

0% packet loss:



```
ron@ron-VirtualBox:~/Documents/Network3-main$ python3 Receiver.py
Got a connection from ('127.0.0.1', 56922)
First file has been received
Authentication has been sent.
Changing CC algoritem to reno.
Second file has been received
First file has been received
Authentication has been sent.
Changing CC algoritem to reno.
Second file has been received
First file has been received
Authentication has been sent.
Changing CC algoritem to reno.
Second file has been received
First file has been received
Authentication has been sent.
Changing CC algoritem to reno.
Second file has been received
First file has been received
Authentication has been sent.
Changing CC algoritem to reno.
Second file has been received
First file has been received
Authentication has been sent.
#####
#####RESULTS#####
File-number      Cubic      Reno
seq = 0 cubic = 0.008491992950439453 reno = 0.08369874954223633
seq = 1 cubic = 0.008529901504516602 reno = 0.05398750305175781
seq = 2 cubic = 0.004377126693725586 reno = 0.05759000778198242
seq = 3 cubic = 0.004558086395263672 reno = 0.04334306716918945
seq = 4 cubic = 0.005424022674560547 reno = 0.04443550109863281
The avarage time for cubic is 0.006276226043701172
The avarage time for reno is 0.05661096572875977
ron@ron-VirtualBox:~/Documents/Network3-main$ 
```

```
ron@ron-VirtualBox:~/Documents/Network3-main$ python3 Sender.py
First half has been sent.
Authentication match.
Second file has been sent.
Send file again? (y/n)y
First half has been sent.
Authentication match.
Second file has been sent.
Send file again? (y/n)y
First half has been sent.
Authentication match.
Second file has been sent.
Send file again? (y/n)y
First half has been sent.
Authentication match.
Second file has been sent.
Send file again? (y/n)y
First half has been sent.
Authentication match.
Second file has been sent.
Send file again? (y/n)y
First half has been sent.
Authentication match.
Second file has been sent.
Send file again? (y/n)y
First half has been sent.
Authentication match.
Second file has been sent.
Send file again? (y/n)y
First half has been sent.
Authentication match.
Second file has been sent.
Send file again? (y/n)n
Exiting...
ron@ron-VirtualBox:~/Documents/Network3-main$ 
```



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	TCP	74	56922 → 9999 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TStamp=2329662852 TSectr=0 WS=
2	0.000017781	127.0.0.1	127.0.0.1	TCP	74	9999 → 56922 [SYN] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM TStamp=2329662852 TSectr=0 WS=
3	0.000032709	127.0.0.1	127.0.0.1	TCP	66	56922 → 9999 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TStamp=2329662852 TSectr=2329662852
4	0.0000881589	127.0.0.1	127.0.0.1	TCP	8258	56922 → 9999 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=8192 TStamp=2329662853 TSectr=2329662852
5	0.0000934993	127.0.0.1	127.0.0.1	TCP	66	9999 → 56922 [ACK] Seq=1 Ack=8193 Win=60800 Len=0 TStamp=2329662853 TSectr=2329662853
6	0.0000998478	127.0.0.1	127.0.0.1	TCP	8258	56922 → 9999 [PSH, ACK] Seq=8193 Ack=1 Win=65536 Len=8192 TStamp=2329662853 TSectr=2329662853
7	0.0001004660	127.0.0.1	127.0.0.1	TCP	66	9999 → 56922 [ACK] Seq=1 Ack=16385 Win=60800 Len=0 TStamp=2329662853 TSectr=2329662853
8	0.0001039142	127.0.0.1	127.0.0.1	TCP	8258	56922 → 9999 [PSH, ACK] Seq=16385 Ack=1 Win=65536 Len=8192 TStamp=2329662853 TSectr=2329662
9	0.0001093515	127.0.0.1	127.0.0.1	TCP	66	9999 → 56922 [ACK] Seq=1 Ack=24577 Win=56704 Len=0 TStamp=2329662853 TSectr=2329662853
10	0.001119469	127.0.0.1	127.0.0.1	TCP	8258	56922 → 9999 [PSH, ACK] Seq=1 Ack=24577 Win=56704 Len=8192 TStamp=2329662853 TSectr=2329662
11	0.001167243	127.0.0.1	127.0.0.1	TCP	32807	56922 → 9999 [ACK] Seq=32769 Ack=1 Win=65536 Len=32741 TStamp=2329662853 TSectr=2329662853
12	0.003092683	127.0.0.1	127.0.0.1	TCP	66	9999 → 56922 [ACK] Seq=1 Ack=65510 Win=36352 Len=0 TStamp=2329662855 TSectr=2329662853
13	0.003126142	127.0.0.1	127.0.0.1	TCP	32807	56922 → 9999 [ACK] Seq=65510 Ack=1 Win=65536 Len=32741 TStamp=2329662855 TSectr=2329662855
14	0.003525423	127.0.0.1	127.0.0.1	TCP	66	9999 → 56922 [ACK] Seq=1 Ack=98251 Win=48640 Len=0 TStamp=2329662856 TSectr=2329662855
15	0.003589687	127.0.0.1	127.0.0.1	TCP	32807	56922 → 9999 [ACK] Seq=98251 Ack=1 Win=65536 Len=32741 TStamp=2329662856 TSectr=2329662856
16	0.003946100	127.0.0.1	127.0.0.1	TCP	66	9999 → 56922 [ACK] Seq=1 Ack=130992 Win=65536 Len=0 TStamp=2329662856 TSectr=2329662856
17	0.004107301	127.0.0.1	127.0.0.1	TCP	32807	56922 → 9999 [ACK] Seq=130992 Ack=1 Win=65536 Len=32741 TStamp=2329662856 TSectr=2329662856
18	0.004113477	127.0.0.1	127.0.0.1	TCP	32807	56922 → 9999 [ACK] Seq=163733 Ack=1 Win=65536 Len=32741 TStamp=2329662856 TSectr=2329662856
19	0.004121945	127.0.0.1	127.0.0.1	TCP	66	9999 → 56922 [ACK] Seq=1 Ack=163733 Win=196480 Len=0 TStamp=2329662856 TSectr=2329662856
20	0.004207942	127.0.0.1	127.0.0.1	TCP	32807	56922 → 9999 [ACK] Seq=196474 Ack=1 Win=65536 Len=32741 TStamp=2329662856 TSectr=2329662856
21	0.004219734	127.0.0.1	127.0.0.1	TCP	32807	56922 → 9999 [ACK] Seq=229215 Ack=1 Win=65536 Len=32741 TStamp=2329662856 TSectr=2329662856

We can see that we sent the file 5 times with 0% packet loss .
The result is that cubic is faster compare to reno.

10% packet loss:

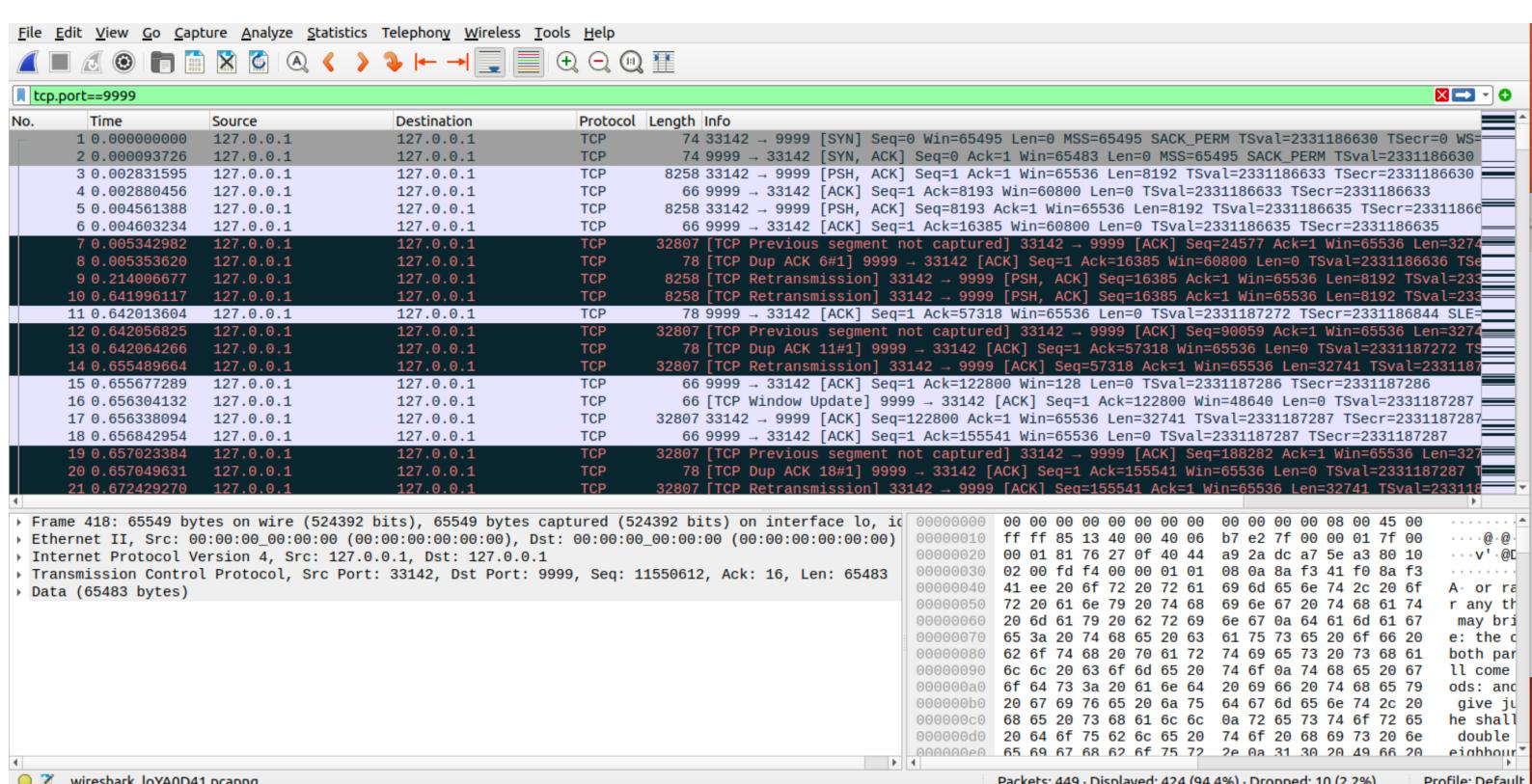
```
ron@ron-VirtualBox:~/Documents/Network3-main$ sudo tc qdisc change dev lo root netem loss  
10%  
ron@ron-VirtualBox:~/Documents/Network3-main$ python3 Sender.py  
First half has been sent.  
Authentication match.  
Second file has been sent.  
Send file again? (y/n)  
First half has been sent.  
Authentication match.  
Second file has been sent.  
Send file again? (y/n)  
First half has been sent.  
Authentication match.  
Second file has been sent.  
Send file again? (y/n)  
First half has been sent.  
Authentication match.  
Second file has been sent.  
Send file again? (y/n)  
First half has been sent.  
Authentication match.  
Second file has been sent.  
Send file again? (y/n)  
Exiting...  
ron@ron-VirtualBox:~/Documents/Network3-main$
```

We can see that we sent the file 5 times with 10% packet loss . The result is that cubic is slower compare to reno.

15% packet loss:

```
ron@ron-VirtualBox:~/Documents/Network3-matt$ python3 Receiver.py
Got a connection from ('127.0.0.1', 33142)
First file has been received
Authentication has been sent.
Changing CC algorithem to reno.
Second file has been received
First file has been received
Authentication has been sent.
Changing CC algorithem to reno.
Second file has been received
First file has been received
Authentication has been sent.
Changing CC algorithem to reno.
Second file has been received
First file has been received
Authentication has been sent.
Changing CC algorithem to reno.
Second file has been received
First file has been received
Authentication has been sent.
Changing CC algorithem to reno.
Second file has been received
First file has been received
Authentication has been sent.
Changing CC algorithem to reno.
Second file has been received
#####
#####RESULTS#####
File-number      Cubic       Reno
seq = 0 cubic = 1.057706594467163 reno = 0.2571902275085449
seq = 1 cubic = 0.5041625499725342 reno = 0.30411434173583984
seq = 2 cubic = 1.6998112201696674 reno = 0.03604865074157715
seq = 3 cubic = 0.2108166217803955 reno = 0.16162848472595215
seq = 4 cubic = 0.5092949867248535 reno = 0.0672312195800781
The avarge time for cubic is 0.7963583946228028
The avarge time for reno is 0.16524276733398438
```

```
ron@ron-VirtualBox:~/Documents/Network3-main$ sudo tc qdisc change dev lo root netem loss  
15%  
ron@ron-VirtualBox:~/Documents/Network3-main$ python3 Sender.py  
First half has been sent.  
Authentication match.  
Second file has been sent.  
Send file again? (y/n)y  
First half has been sent.  
Authentication match.  
Second file has been sent.  
Send file again? (y/n)y  
First half has been sent.  
Authentication match.  
Second file has been sent.  
Send file again? (y/n)y  
First half has been sent.  
Authentication match.  
Second file has been sent.  
Send file again? (y/n)y  
First half has been sent.  
Authentication match.  
Second file has been sent.  
Send file again? (y/n)n  
Exiting...  
ron@ron-VirtualBox:~/Documents/Network3-main$ 
```



We can see that we sent the file 5 times with 15% packet loss. The result is that cubic is slower compared to reno.

20% packet loss:

```
ron@ron-VirtualBox:~/Documents/Network3-main$ python3 Receiver.py
Got a connection from ('127.0.0.1', 40404)
First file has been received
Authentication has been sent.
Changing CC algorithem to reno.
Second file has been received
First file has been received
Authentication has been sent.
Changing CC algorithem to reno.
Second file has been received
First file has been received
Authentication has been sent.
Changing CC algorithem to reno.
Second file has been received
First file has been received
Authentication has been sent.
Changing CC algorithem to reno.
Second file has been received
First file has been received
Authentication has been sent.
Changing CC algorithem to reno.
Second file has been received
First file has been received
Authentication has been sent.
Changing CC algorithem to reno.
Second file has been received
First file has been received
Authentication has been sent.
Changing CC algorithem to reno.
Second file has been received
First file has been received
Authentication has been sent.
#####
#####RESULTS#####
File-number      Cubic      Reno
seq = 0 cubic = 1.141979455947876 reno = 1.2220096588134766
seq = 1 cubic = 10.372520923614502 reno = 9.2096266746521
seq = 2 cubic = 5.24347710609436 reno = 0.06096363067626953
seq = 3 cubic = 0.21830487251281738 reno = 0.30440711975097656
seq = 4 cubic = 0.491147518157959 reno = 6.408109426498413
The avarage time for cubic is 3.4934859752655028
The avarage time for reno is 3.4410233020782472
ron@ron-VirtualBox:~/Documents/Network3-main$
```

```
ron@ron-VirtualBox:~/Documents/Network3-main$ sudo tc qdisc change dev lo root netem loss
20%
ron@ron-VirtualBox:~/Documents/Network3-main$ python3 Sender.py
First half has been sent.
Authentication match.
Second file has been sent.
Send file again? (y/n)y
First half has been sent.
Authentication match.
Second file has been sent.
Send file again? (y/n)y
First half has been sent.
Authentication match.
Second file has been sent.
Send file again? (y/n)y
First half has been sent.
Authentication match.
Second file has been sent.
Send file again? (y/n)y
First half has been sent.
Authentication match.
Second file has been sent.
Send file again? (y/n)n
Exiting...
ron@ron-VirtualBox:~/Documents/Network3-main$
```

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000000000	127.0.0.1	127.0.0.1	TCP	74	40404 → 9999 [SYN] Seq=0 Win=65495 MSS=65495 SACK_PERM TStamp=2332283750 TSecr=0 WS=
2	0.000660669	127.0.0.1	127.0.0.1	TCP	74	9999 → 40404 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM TStamp=2332283751
3	0.001515862	127.0.0.1	127.0.0.1	TCP	66	40404 → 9999 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TStamp=2332283752 TSecr=2332283751
4	0.210394642	127.0.0.1	127.0.0.1	TCP	32807	[TCP Previous segment not captured] 40404 → 9999 [ACK] Seq=8193 Ack=1 Win=65536 Len=32741
5	0.210428223	127.0.0.1	127.0.0.1	TCP	78	[TCP Window Update] 9999 → 40404 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TStamp=2332283962 TSecr=2332283962
6	0.210450382	127.0.0.1	127.0.0.1	TCP	8258	[TCP Out-of-order] 40404 → 9999 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=8192 TStamp=233228396
7	0.420199873	127.0.0.1	127.0.0.1	TCP	8258	[TCP Retransmission] 40404 → 9999 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=8192 TStamp=2332284
8	0.420219718	127.0.0.1	127.0.0.1	TCP	78	9999 → 40404 [ACK] Seq=1 Ack=40934 Win=65536 Len=0 TStamp=2332284171 TSecr=2332283962 SLE=
9	0.420272763	127.0.0.1	127.0.0.1	TCP	32807	40404 → 9999 [ACK] Seq=40934 Ack=1 Win=65536 Len=32741 TStamp=2332284171 TSecr=2332284171
10	0.420421158	127.0.0.1	127.0.0.1	TCP	32807	40404 → 9999 [ACK] Seq=73675 Ack=1 Win=65536 Len=32741 TStamp=2332284171 TSecr=2332284171
11	0.420469479	127.0.0.1	127.0.0.1	TCP	66	9999 → 40404 [ACK] Seq=1 Ack=73675 Win=48640 Len=0 TStamp=2332284171 TSecr=2332284171
12	0.420480041	127.0.0.1	127.0.0.1	TCP	66	9999 → 40404 [ACK] Seq=1 Ack=106416 Win=16000 Len=0 TStamp=2332284172 TSecr=2332284171
13	0.421496604	127.0.0.1	127.0.0.1	TCP	66	[TCP Window Update] 9999 → 40404 [ACK] Seq=1 Ack=106416 Win=48640 Len=0 TStamp=2332284173
14	0.421530118	127.0.0.1	127.0.0.1	TCP	32807	40404 → 9999 [ACK] Seq=106416 Ack=1 Win=65536 Len=32741 TStamp=2332284173 TSecr=2332284173
15	0.422503120	127.0.0.1	127.0.0.1	TCP	66	9999 → 40404 [ACK] Seq=1 Ack=139157 Win=65536 Len=0 TStamp=2332284174 TSecr=2332284173
16	0.422560145	127.0.0.1	127.0.0.1	TCP	32807	40404 → 9999 [ACK] Seq=139157 Ack=1 Win=65536 Len=32741 TStamp=2332284174 TSecr=2332284174
17	0.422567731	127.0.0.1	127.0.0.1	TCP	32807	40404 → 9999 [ACK] Seq=171898 Ack=1 Win=65536 Len=32741 TStamp=2332284174 TSecr=2332284174
18	0.422644554	127.0.0.1	127.0.0.1	TCP	66	9999 → 40404 [ACK] Seq=1 Ack=204639 Win=128 Len=0 TStamp=2332284174 TSecr=2332284174
19	0.646917558	127.0.0.1	127.0.0.1	TCP	194	[TCP Window Full] 40404 → 9999 [PSH, ACK] Seq=204639 Ack=1 Win=65536 Len=128 TStamp=2332284174
20	0.875205260	127.0.0.1	127.0.0.1	TCP	194	[TCP Window Full] [TCP Retransmission] 40404 → 9999 [PSH, ACK] Seq=204639 Ack=1 Win=65536
21	0.875225016	127.0.0.1	127.0.0.1	TCP	78	9999 → 40404 [ACK] Seq=1 Ack=1 Win=40404 Len=196608 TStamp=2332284626 TSecr=2332284626

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface lo, id 0
 Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
 Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
 Transmission Control Protocol, Src Port: 40404, Dst Port: 9999, Seq: 0, Len: 0

0000	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 00 00 00 08 00 45 00
0010	00 3c ba 0a 40 00 40 06	82 af 7f 00 00 01 7f 00	<..@..
0020	00 01 9d d4 27 0f 69 e4	3b c0 00 00 00 00 a0 02	..1..;
0030	ff d7 fe 30 00 00 02 04	ff d7 04 02 08 0a 8b 03	0..
0040	d3 66 00 00 00 00 01 03	03 07	f...:..

We can see that we sent the file 5 times with 20% packet loss .
 The result is that cubic is slower compare to reno.

Conclusion:

According to the results above we can see that when we running the receiver and the sender with 0% packets loss the cubic was faster, but when we added the precent to 10,15,20 the reno is faster.

We search on the Internet about the two algorithms and we read that cubic are more effective when the conditions are favorable however the reno is better suited for older networks that have lower bandwidth and shorter delays. The result are matched to our conclusion.

Explain about function in the code:

server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1):

This function sets sockets option for the server to reuse the local address when binding to a new socket.

server.bind(SERVER_ADDR):

This function assigns a specific network address to the server for communication.

server.listen(NUM_CONNECTIONS):

This function sets the server to listen for incoming client connections with the maximum number specified.

server.accept():

This function is waiting until a client is connecting.

sendall():

This function sends a sequence of bytes to the other end of the socket connection, making sure that all data is transmitted.

client.setsockopt(socket.IPPROTO_TCP, socket.TCP_CONGESTION, CC):

This function sets the CC algorithm for a TCP connection established by the client to the CC algorithm by choice.