COMP 8005: Final Project

Port Forwarding

Kevin Lo, A00952922
4-4-2021

Table of Contents

1 Introduction

This project is to create a program that port forwards any IP + port pair to another IP + port pair and to test its functionality such as performance and functions.

2 How to Use

Input the IP + port combinations to forward in **forward.txt** and interfaces to use for port forwarding in **forward.txt** then run the **portforward.py** on the Linux machine to be used as a port forwarder by using iptables rules to forward the rules to the computer.

Eg. forward.txt

srcIP srcDport dstIP dstDport

forward.txt

```
1   192.168.1.253 1001 192.168.1.252 22
2   192.168.1.253 1002 192.168.1.252 80
3   192.168.1.253 1003 192.168.1.252 8000
```

Forwards packets coming from 192.168.1.253 on dport 1001 to 192.168.1.252 dport 22
Forwards packets coming from 192.168.1.253 on dport 1002 to 192.168.1.252 dport 80
Forwards packets coming from 192.168.1.253 on dport 1003 to 192.168.1.252 dport 8000

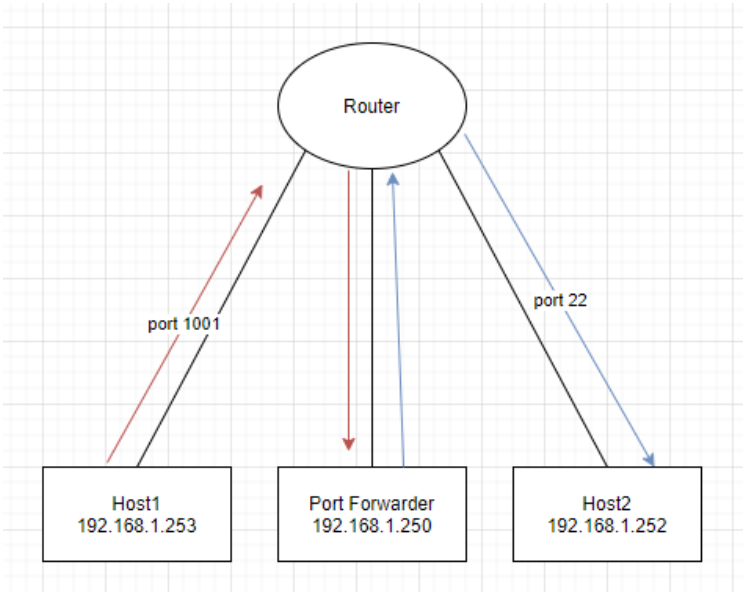
Eg. interfaces.txt

interfaces.txt

```
1   wlp2s0
```

Overwrites any packets coming out my wlp2s0 NIC interface as the port forward's IP, this is required for the port forwarder to work properly. You may need to use more than one interface if it is required to port forward to more than one network.

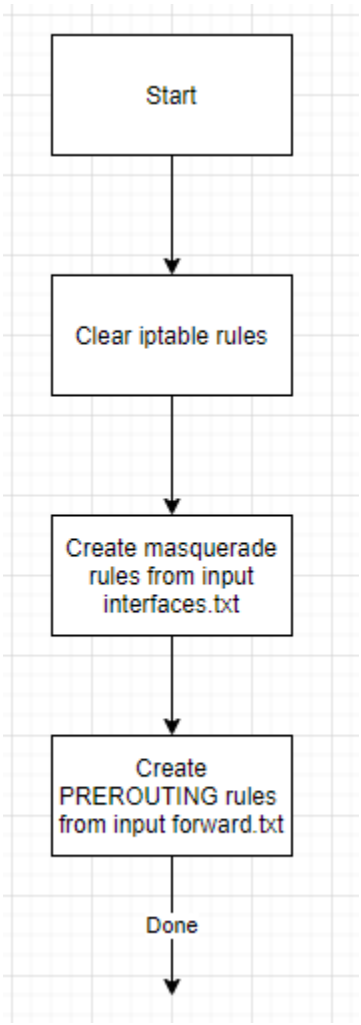
Example of how going from host 192.168.1.253:1001 to get forwarded to 192.168.1.252:22 by sending it to 192.168.1.250 on dport 1001.



3 Operating System

At minimum, the computer used as the port forwarder must be using any form of Linux distro that has iptables to use for port forwarding because this program creates iptables rules to do port forwarding.

4 Design



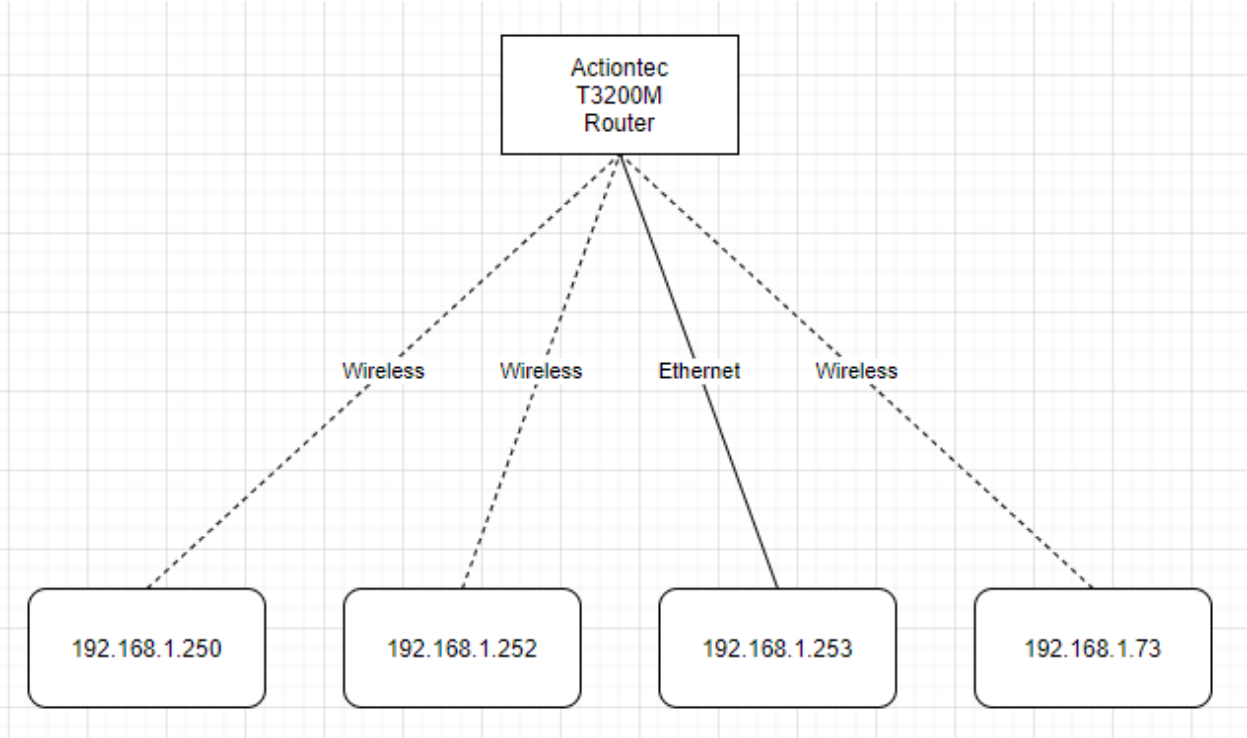
5 Pseudocode

```
Clear all iptables rules
read interface.txt
        for line in interface.txt
                run (“iptables – A POSTROUTING -t nat -o “ + line + “ -j MASQUERADE”)
read forward.txt
        for line in forward.txt
                run(“iptables -A PREROUTING -t nat -s “ + srcIP + “ p tcp – dport “ + srcDport + “ -j DNAT
–to-destination “ + dstIP + “:” + dstDport)
```

6 Testbench

There is a list of 4 computers as hardware I use for this project to test the functionality of my port forwarder.

- 192.168.1.250 (i7 6500U 12GB RAM Laptop)
- 192.168.1.252 (Raspberry Pi 4)
- 192.168.1.253 (Ryzen 7 3800x 32GB RAM Desktop)
- 192.168.1.73 (i7 4790k 32GB RAM Desktop)



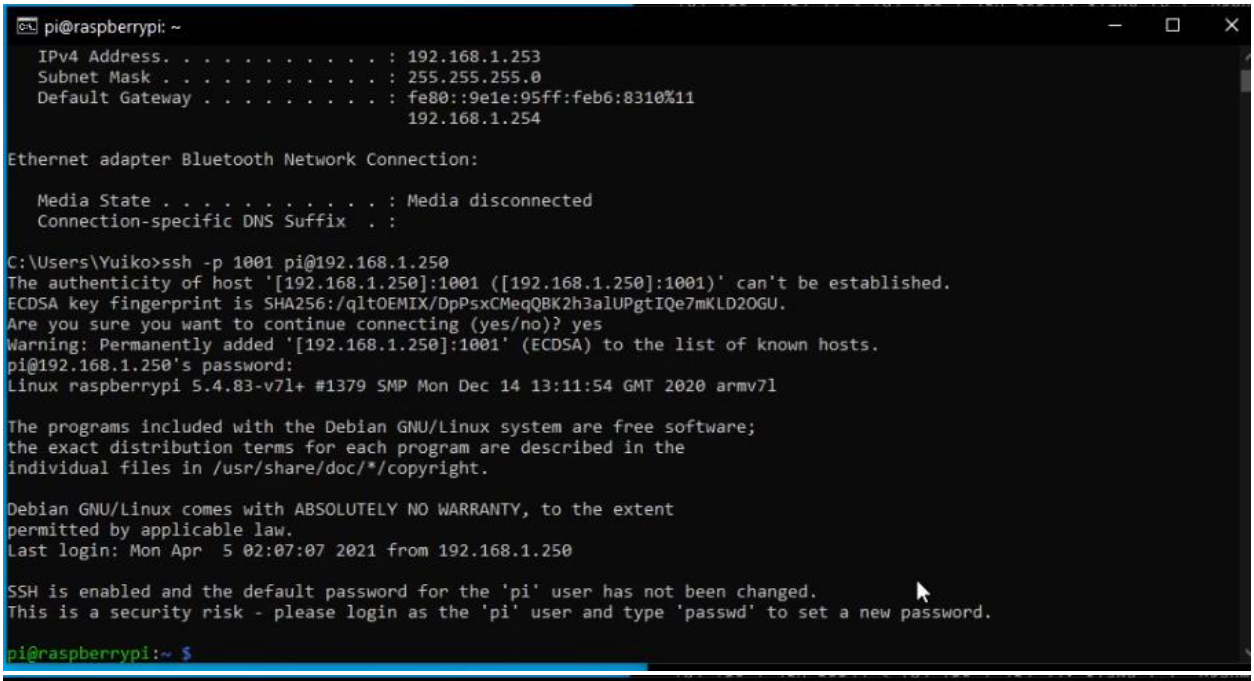
7 Tests

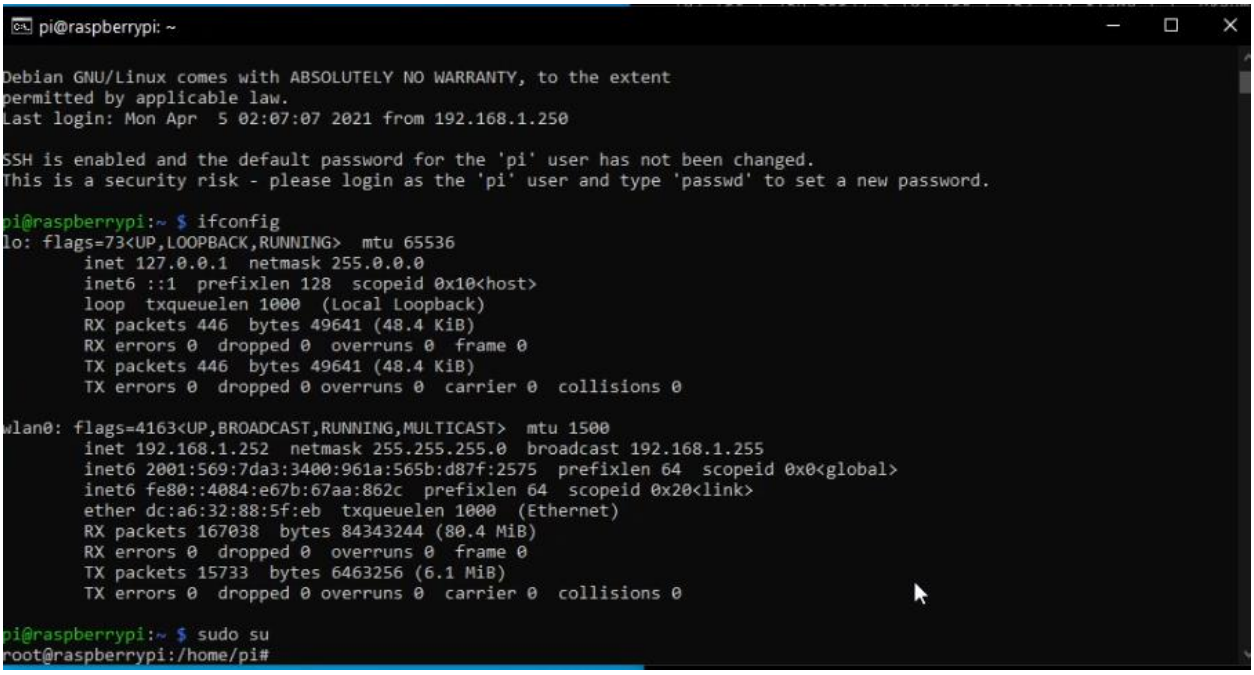
Tests were done with mostly the same computers however, for test 8 I used an RPI4 as port forwarder because it was unable to run my Epoll server code.

Rule #	Test Description	Tool Used	Expected Result	Pass/Fail
1	Send 192.168.1.253 dport 1001 to the forwarder to forward to 192.168.1.252 dport 22 for SSH. ***Client: 192.168.1.253*** ***Forwarder: 192.168.1.250*** ***Server: 192.168.1.252***	SSH	The rule should forward the packet to 192.168.1.252 and behave like regular SSH	Pass
2	Send 192.168.1.253 dport 1002 to the forwarder to forward to 192.168.1.252 dport 80 for HTML. ***Client: 192.168.1.253*** ***Forwarder: 192.168.1.250*** ***Server: 192.168.1.252***	IE Explorer	The rule should forward the packet to 192.168.1.252 and show the default apache page on 192.168.1.252	Pass
3	Send 192.168.1.253 dport 1003 to the forwarder to forward to 192.168.1.252 dport 8000 for echo server testing. ***Client: 192.168.1.253*** ***Forwarder: 192.168.1.250*** ***Server: 192.168.1.252***	Echo Server Client	The rule should forward the packet 192.168.1.252 and return a message to 192.168.1.253	Pass
4	100 client load using epoll server on dport 1004>8000 ***Client: 192.168.1.253*** ***Forwarder: 192.168.1.250*** ***Server: 192.168.1.73***	Epoll server	It should function normally but slightly slower than without the forwarder	Pass
5	500 client load using epoll server on dport 1004>8000 ***Client: 192.168.1.253*** ***Forwarder: 192.168.1.250*** ***Server: 192.168.1.73***	Epoll server	It should function normally but slightly slower than without the forwarder	Pass
6	1000 client load using epoll server on dport 1004>8000 ***Client: 192.168.1.253*** ***Forwarder: 192.168.1.250*** ***Server: 192.168.1.73***	Epoll server	It should function normally but slightly slower than without the forwarder	Pass
7	1500 client load using epoll server on dport 1004>8000 ***Client: 192.168.1.253*** ***Forwarder: 192.168.1.250*** ***Server: 192.168.1.73***	Epoll server	It should function normally but slightly slower than without the forwarder	Fail
8	100+100 client two way traffic ***Host1: 192.168.1.250*** ***Forwarder: 192.168.1.252*** ***Host2: 192.168.1.73***	Epoll server	It should be work but might be slow because RPI4 forwarder	Pass

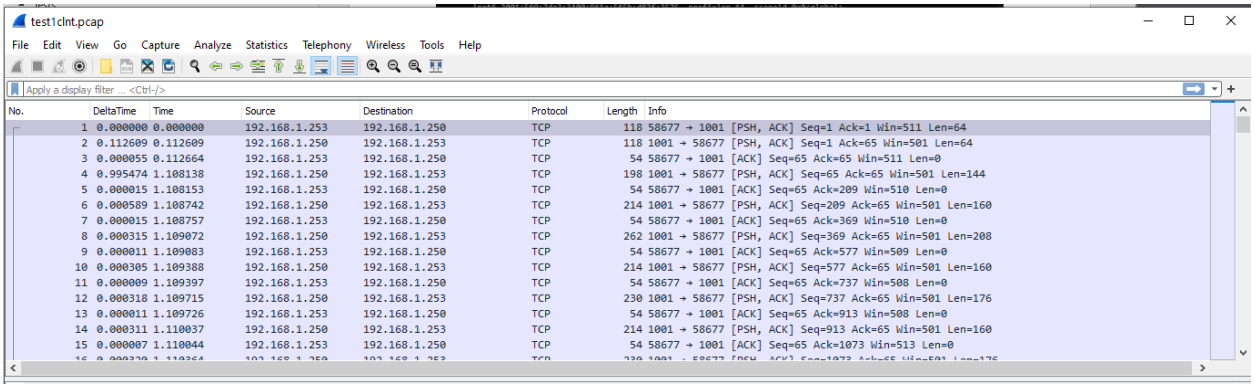
7.1 Test 1 SSH

I was able to successfully be forwarded to 192.168.1.252 dport 22 by SSH 192.168.1.250 dport 1001. The **forward.txt** rule used to do this is **192.168.1.253 1001 192.168.1.252 22** which means to forward any packets from 192.168.1.253 going to my port forwarder on port 1001 will then be port forwarded to 192.168.1.252 22. Both client and server will communicate to 192.168.1.250 as the intermediary.

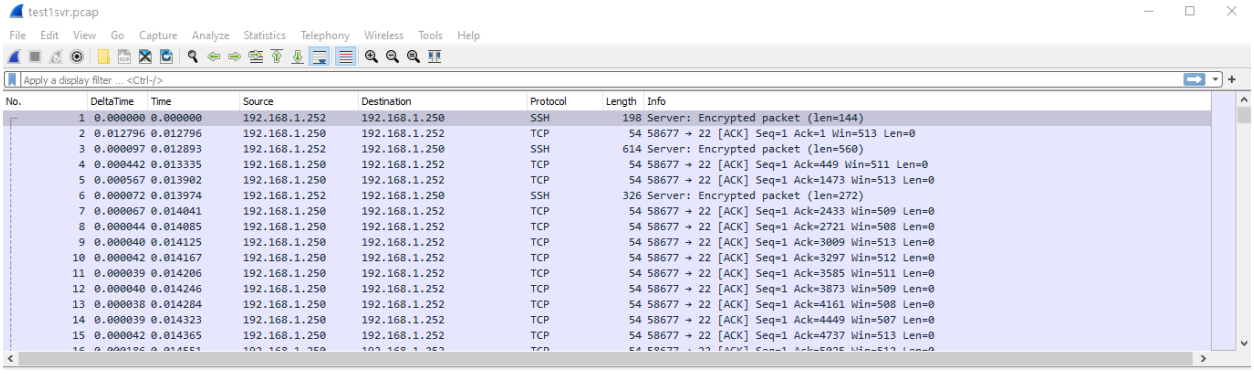




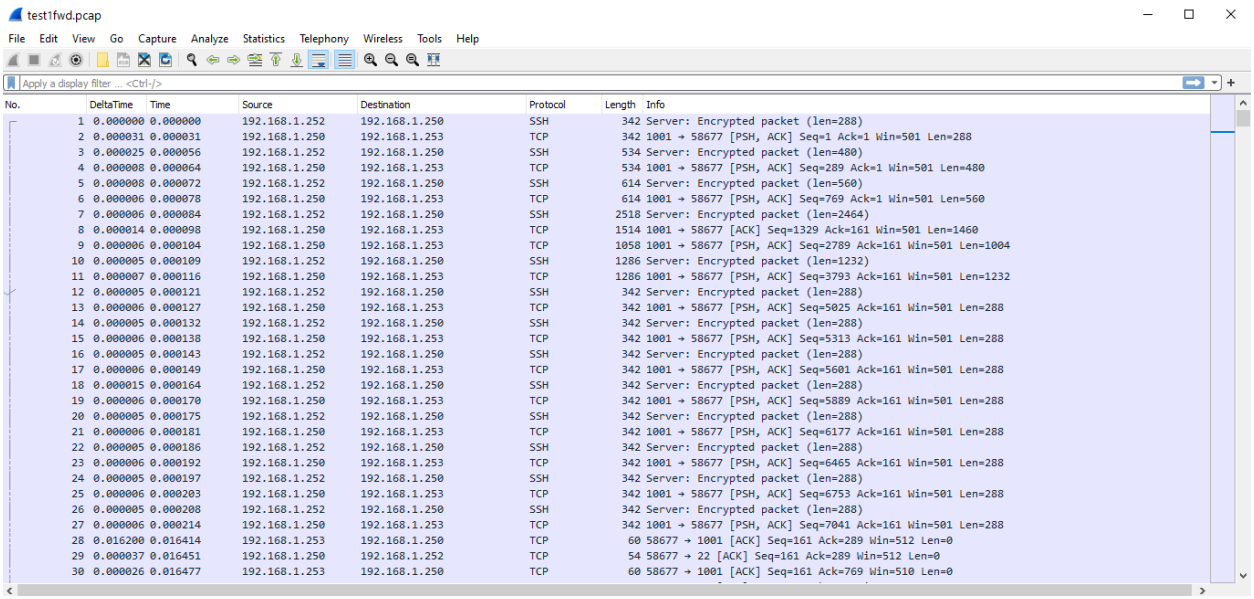
Client



Server



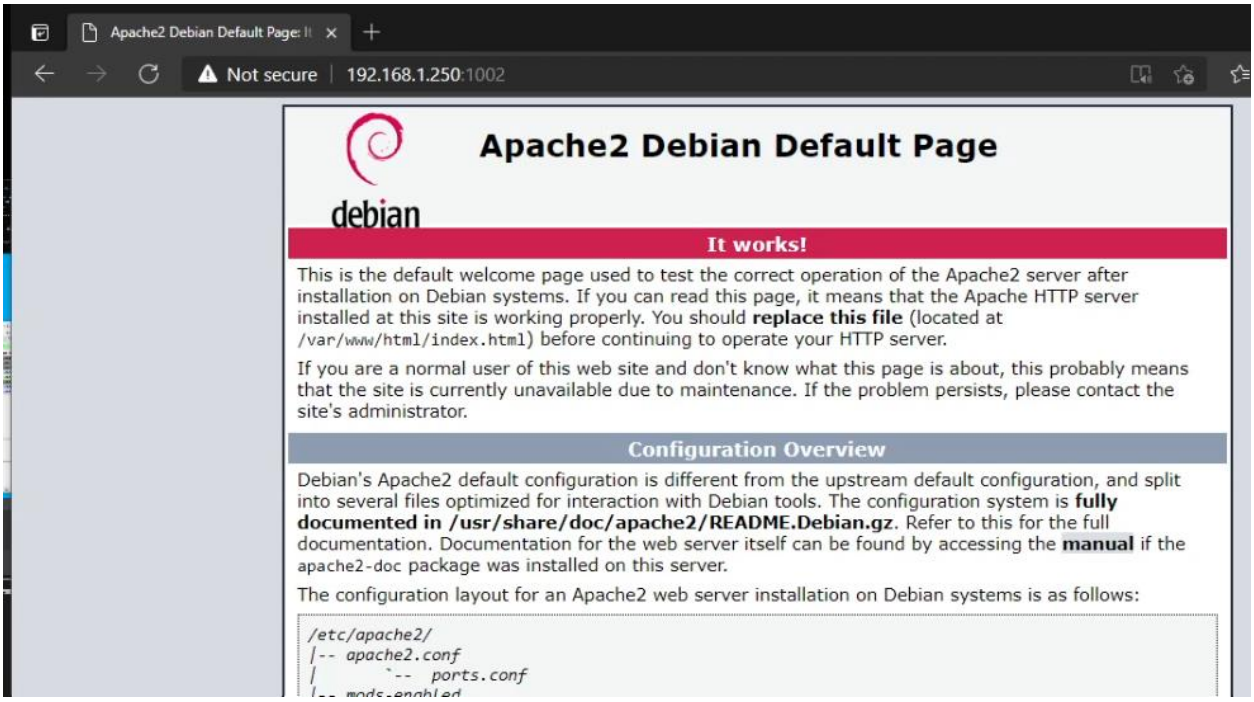
Forwarder



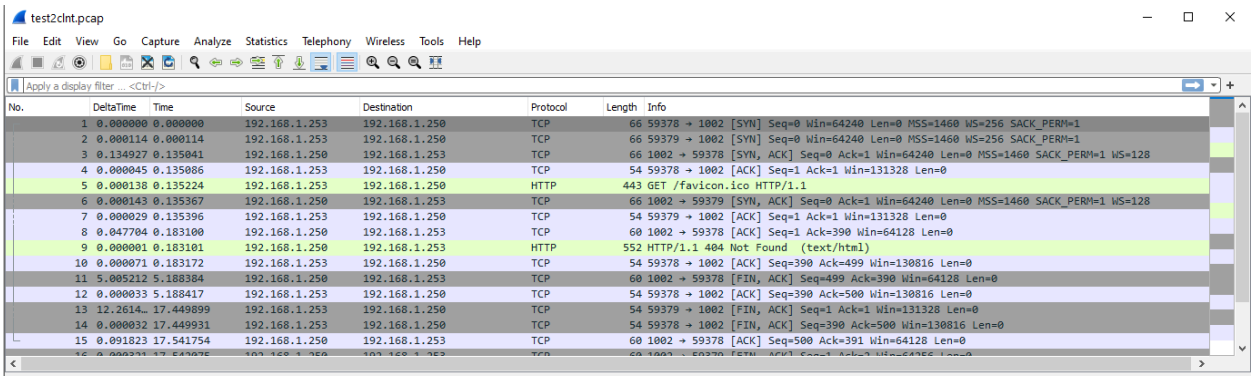
As shown by the packet captures the forwarder is the one responsible for manging communications between both client and server by passing packets to eachother

7.2 Test 2 HTTP

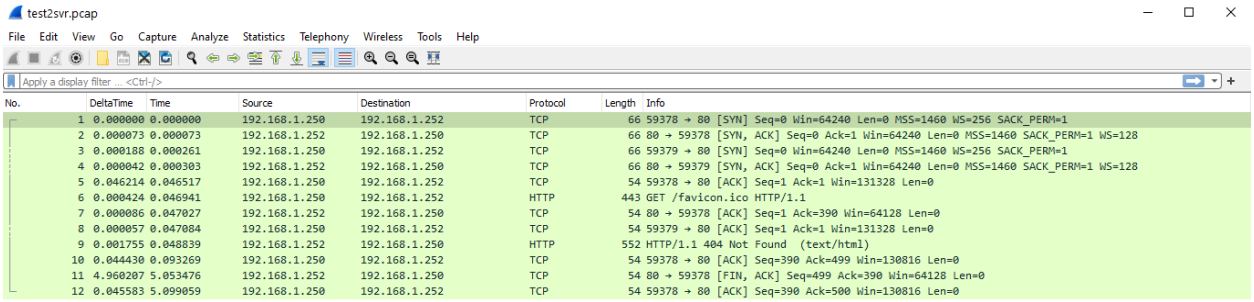
I was able to successfully be forwarded to 192.168.1.252 dport 80 by HTML 192.168.1.250 dport 1002. The **forward.txt** rule used to do this is **192.168.1.253 1002 192.168.1.252 80** which means to forward any packets from 192.168.1.253 going to my port forwarder on port 1002 will then be port forwarded to 192.168.1.252 80. Both client and server will communicate to 192.168.1.250 as the intermediary.



Client

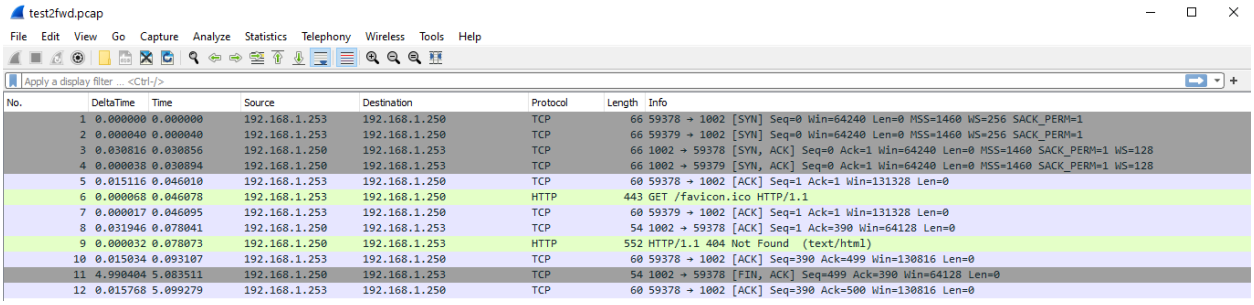


Server



No.	DeltaTime	Time	Source	Destination	Protocol	Length	Info
1	0.000000	0.000000	192.168.1.250	192.168.1.252	TCP	66	59378 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000073	0.000073	192.168.1.252	192.168.1.250	TCP	66	80 → 59378 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000188	0.000261	192.168.1.250	192.168.1.252	TCP	66	59379 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
4	0.000042	0.000303	192.168.1.252	192.168.1.250	TCP	66	80 → 59379 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
5	0.046214	0.046517	192.168.1.250	192.168.1.252	TCP	54	59378 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
6	0.000424	0.046941	192.168.1.250	192.168.1.252	HTTP	443	GET /favicon.ico HTTP/1.1
7	0.000086	0.047027	192.168.1.252	192.168.1.250	TCP	54	80 → 59378 [ACK] Seq=1 Ack=390 Win=64128 Len=0
8	0.000057	0.047084	192.168.1.250	192.168.1.252	TCP	54	59379 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
9	0.001755	0.048839	192.168.1.252	192.168.1.250	HTTP	552	HTTP/1.1 404 Not Found (text/html)
10	0.044430	0.093269	192.168.1.250	192.168.1.252	TCP	54	59378 → 80 [ACK] Seq=390 Ack=499 Win=130816 Len=0
11	4.960207	5.053476	192.168.1.252	192.168.1.250	TCP	54	80 → 59378 [FIN, ACK] Seq=499 Ack=390 Win=64128 Len=0
12	0.045583	5.099059	192.168.1.250	192.168.1.252	TCP	54	59378 → 80 [ACK] Seq=390 Ack=500 Win=130816 Len=0

Forwarder



No.	DeltaTime	Time	Source	Destination	Protocol	Length	Info
1	0.000000	0.000000	192.168.1.253	192.168.1.250	TCP	66	59378 → 1002 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000040	0.000040	192.168.1.253	192.168.1.250	TCP	66	59379 → 1002 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
3	0.030816	0.030856	192.168.1.250	192.168.1.253	TCP	66	1002 → 59378 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
4	0.000038	0.030894	192.168.1.250	192.168.1.253	TCP	66	1002 → 59379 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
5	0.015116	0.046010	192.168.1.253	192.168.1.250	TCP	60	59378 → 1002 [ACK] Seq=1 Ack=1 Win=131328 Len=0
6	0.000068	0.046078	192.168.1.253	192.168.1.250	HTTP	443	GET /favicon.ico HTTP/1.1
7	0.000017	0.046095	192.168.1.253	192.168.1.250	TCP	60	59379 → 1002 [ACK] Seq=1 Ack=1 Win=131328 Len=0
8	0.031946	0.078041	192.168.1.250	192.168.1.253	TCP	54	1002 → 59378 [ACK] Seq=1 Ack=390 Win=64128 Len=0
9	0.000032	0.078073	192.168.1.250	192.168.1.253	HTTP	552	HTTP/1.1 404 Not Found (text/html)
10	0.015034	0.093107	192.168.1.253	192.168.1.250	TCP	60	59378 → 1002 [ACK] Seq=390 Ack=499 Win=130816 Len=0
11	4.990404	5.083511	192.168.1.250	192.168.1.253	TCP	54	1002 → 59378 [FIN, ACK] Seq=499 Ack=390 Win=64128 Len=0
12	0.015768	5.099279	192.168.1.253	192.168.1.250	TCP	60	59378 → 1002 [ACK] Seq=390 Ack=500 Win=130816 Len=0

As shown by the packet captures the forwarder is the one responsible for manging communications between both client and server by passing packets to eachother

7.3 Test 3 Echo Server Client

I was able to successfully be forwarded to 192.168.1.252 dport 8000 by HTML 192.168.1.250 dport 1003. The **forward.txt** rule used to do this is **192.168.1.253 1003 192.168.1.252 8000** which means to forward any packets from 192.168.1.253 going to my port forwarder on port 1003 will then be port forwarded to 192.168.1.252 8000. Both client and server will communicate to 192.168.1.250 as the intermediary.

```
E:\Homework\term6\8005\FinalProj\demoecho>python echo-client.py 192.
168.1.250
Received From Server: b'Echo => Hello TCP World'
```

```
pi@raspberrypi: ~
root@raspberrypi:/home/pi/Desktop# dir
echo-server.py  epoll_svr.py  test1svr.pcap  test2svr.pcap
root@raspberrypi:/home/pi/Desktop# python echo-server.py

('Client Connection:', ('192.168.1.250', 59594))
```

Client

test3clnt.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	DeltaTime	Time	Source	Destination	Protocol	Length	Info
1	0.000000	0.000000	192.168.1.253	192.168.1.250	TCP	66	59594 → 1003 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.256926	0.256926	192.168.1.250	192.168.1.253	TCP	66	1003 → 59594 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000081	0.257007	192.168.1.253	192.168.1.250	TCP	54	59594 → 1003 [ACK] Seq=1 Ack=1 Win=131328 Len=0
4	0.000050	0.257057	192.168.1.253	192.168.1.250	TCP	69	59594 → 1003 [PSH, ACK] Seq=1 Ack=1 Win=131328 Len=15
5	0.044916	0.301973	192.168.1.250	192.168.1.253	TCP	60	1003 → 59594 [ACK] Seq=1 Ack=16 Win=64256 Len=0
6	0.000306	0.302279	192.168.1.250	192.168.1.253	TCP	77	1003 → 59594 [PSH, ACK] Seq=1 Ack=16 Win=64256 Len=23
7	0.000027	0.302306	192.168.1.253	192.168.1.250	TCP	54	59594 → 1003 [ACK] Seq=16 Ack=24 Win=131328 Len=0
8	0.000148	0.302454	192.168.1.253	192.168.1.250	TCP	54	59594 → 1003 [FIN, ACK] Seq=16 Ack=24 Win=131328 Len=0
9	0.040229	0.342683	192.168.1.250	192.168.1.253	TCP	60	1003 → 59594 [FIN, ACK] Seq=24 Ack=17 Win=64256 Len=0
10	0.000037	0.342720	192.168.1.253	192.168.1.250	TCP	54	59594 → 1003 [ACK] Seq=17 Ack=25 Win=131328 Len=0

Server

test3svr.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	DeltaTime	Time	Source	Destination	Protocol	Length	Info
1	0.000000	0.000000	192.168.1.250	192.168.1.252	TCP	66	59594 → 8000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000063	0.000063	192.168.1.252	192.168.1.250	TCP	66	8000 → 59594 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.046780	0.046843	192.168.1.250	192.168.1.252	TCP	54	59594 → 8000 [ACK] Seq=1 Ack=1 Win=131328 Len=0
4	0.000080	0.046923	192.168.1.250	192.168.1.252	TCP	69	59594 → 8000 [PSH, ACK] Seq=1 Ack=1 Win=131328 Len=15
5	0.000033	0.046956	192.168.1.252	192.168.1.250	TCP	54	8000 → 59594 [ACK] Seq=1 Ack=16 Win=64256 Len=0
6	0.000411	0.047367	192.168.1.252	192.168.1.250	TCP	77	8000 → 59594 [PSH, ACK] Seq=1 Ack=16 Win=64256 Len=23
7	0.044685	0.092052	192.168.1.250	192.168.1.252	TCP	54	59594 → 8000 [ACK] Seq=16 Ack=24 Win=131328 Len=0
8	0.000737	0.092789	192.168.1.252	192.168.1.250	TCP	54	59594 → 8000 [FIN, ACK] Seq=16 Ack=24 Win=131328 Len=0
9	0.000206	0.092995	192.168.1.252	192.168.1.250	TCP	54	8000 → 59594 [FIN, ACK] Seq=24 Ack=17 Win=64256 Len=0
10	0.039187	0.132182	192.168.1.250	192.168.1.252	TCP	54	59594 → 8000 [ACK] Seq=17 Ack=25 Win=131328 Len=0

Forwarder

test3fwd.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

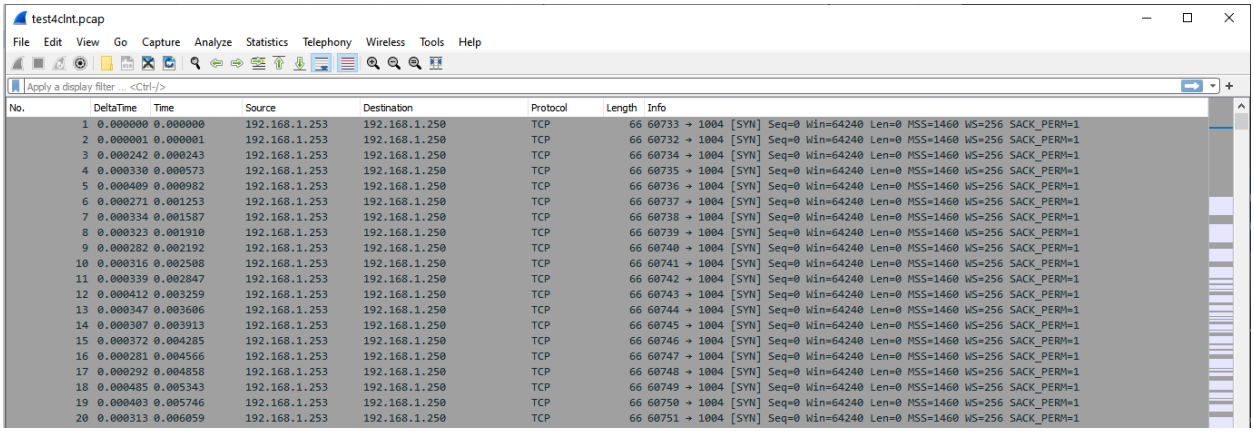
Apply a display filter ... <Ctrl-/>

No.	DeltaTime	Time	Source	Destination	Protocol	Length	Info
1	0.000000	0.000000	192.168.1.250	192.168.1.252	TCP	66	59594 → 8000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.031971	0.031971	192.168.1.252	192.168.1.250	TCP	66	8000 → 59594 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.015751	0.047722	192.168.1.250	192.168.1.252	TCP	54	59594 → 8000 [ACK] Seq=1 Ack=1 Win=131328 Len=0
4	0.000014	0.047736	192.168.1.250	192.168.1.252	TCP	69	59594 → 8000 [PSH, ACK] Seq=1 Ack=1 Win=131328 Len=15
5	0.029868	0.077604	192.168.1.252	192.168.1.250	TCP	54	8000 → 59594 [ACK] Seq=1 Ack=16 Win=64256 Len=0
6	0.000040	0.077644	192.168.1.252	192.168.1.250	TCP	77	8000 → 59594 [PSH, ACK] Seq=1 Ack=16 Win=64256 Len=23
7	0.015669	0.093313	192.168.1.250	192.168.1.252	TCP	54	59594 → 8000 [ACK] Seq=16 Ack=24 Win=131328 Len=0
8	0.000024	0.093337	192.168.1.250	192.168.1.252	TCP	54	59594 → 8000 [FIN, ACK] Seq=16 Ack=24 Win=131328 Len=0
9	0.024846	0.118183	192.168.1.252	192.168.1.250	TCP	54	8000 → 59594 [FIN, ACK] Seq=24 Ack=17 Win=64256 Len=0
10	0.015247	0.133430	192.168.1.250	192.168.1.252	TCP	54	59594 → 8000 [ACK] Seq=17 Ack=25 Win=131328 Len=0

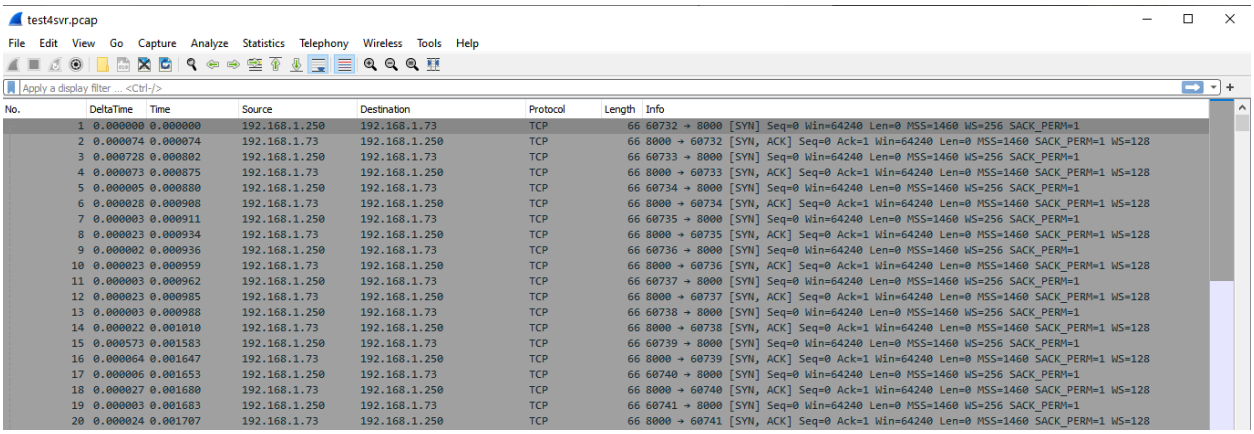
7.4 Test 4 100 Clients

The performance of the port forwarder with 100 clients coming from 192.168.1.253 is relavitely quick and successfully completes the test

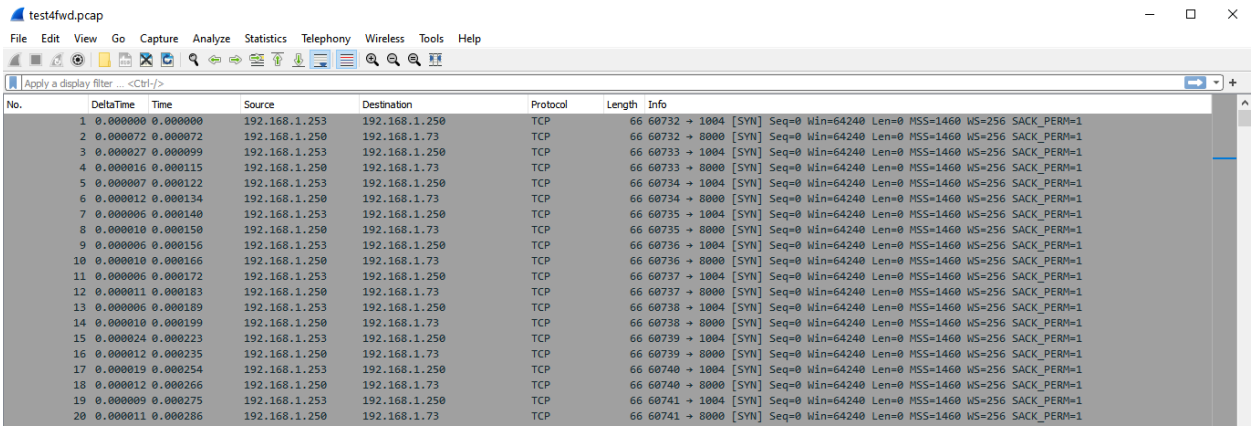
Client

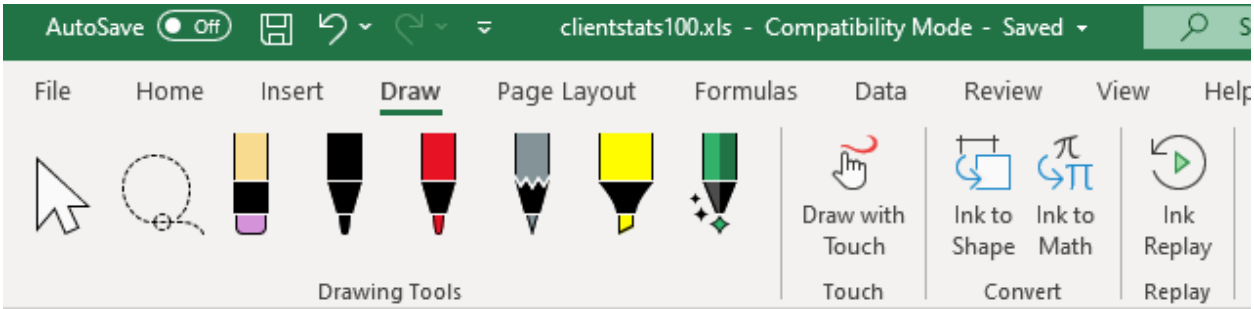


Server



Forwarder





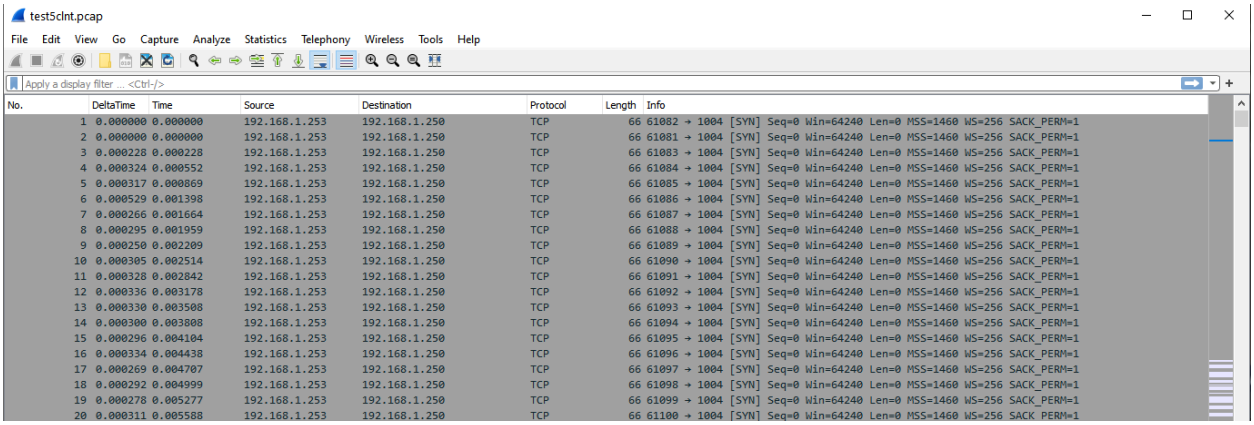
	A	B	C	D	E	F	G	H
1	Thread Nu	Requests	Data Trans	Total Durat	Avg Response		Avg Total Duration	Avg Avg Response
2	0	100	6400	9.150835	0.055895		9.089017622	0.055820329
3	1	100	6400	8.994547	0.055863			
4	2	100	6400	8.675776	0.053789			
5	3	100	6400	9.072159	0.055575			
6	4	100	6400	9.021348	0.055714			
7	5	100	6400	8.925167	0.054117			
8	6	100	6400	9.03405	0.056538			
9	7	100	6400	9.126406	0.056009			
10	8	100	6400	8.94471	0.054776			
11	9	100	6400	9.021348	0.055077			

The average total duration and average response time has increased by almost 3x when compared to having no forwarder

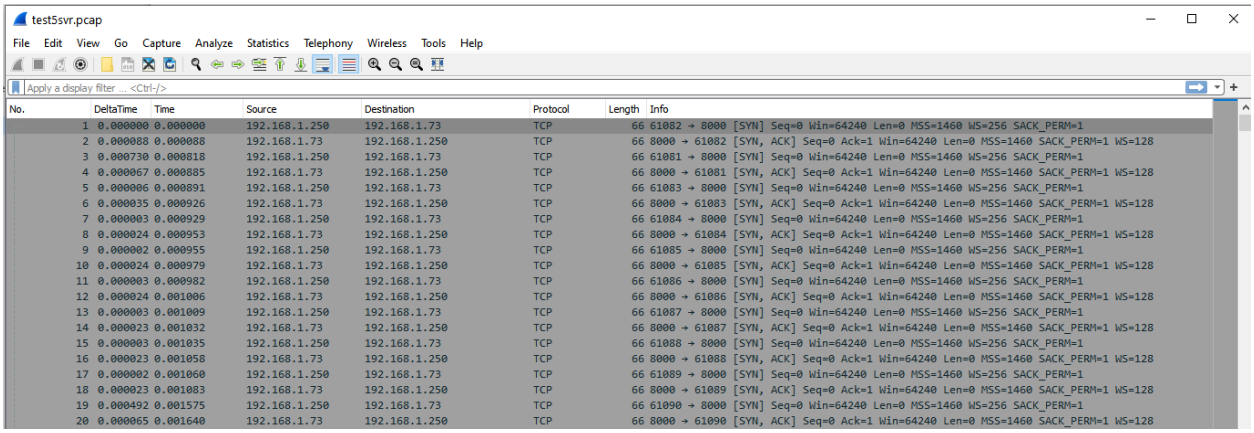
7.5 Test 5 500 Clients

The performance of the port forwarder with 500 clients coming from 192.168.1.253 is slower and successfully completes the test

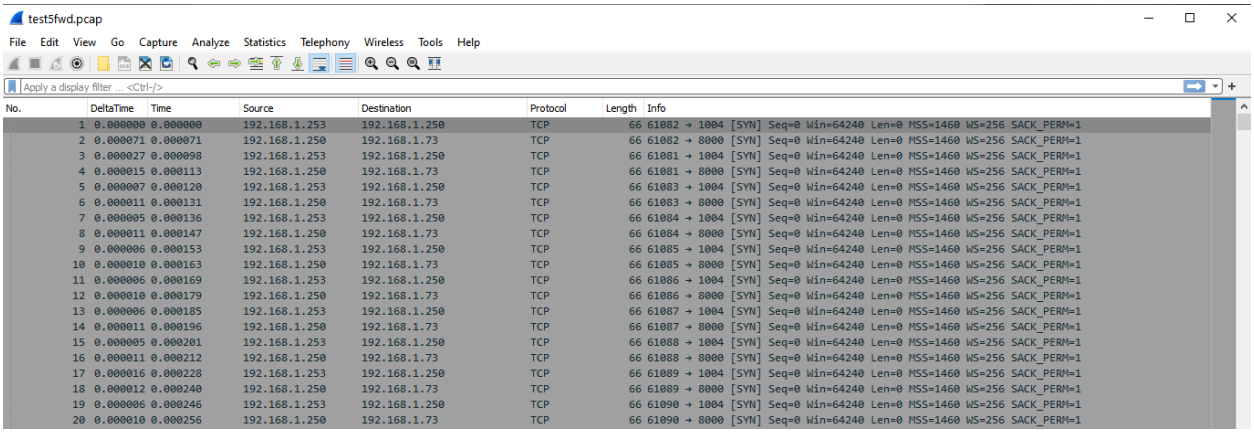
Client

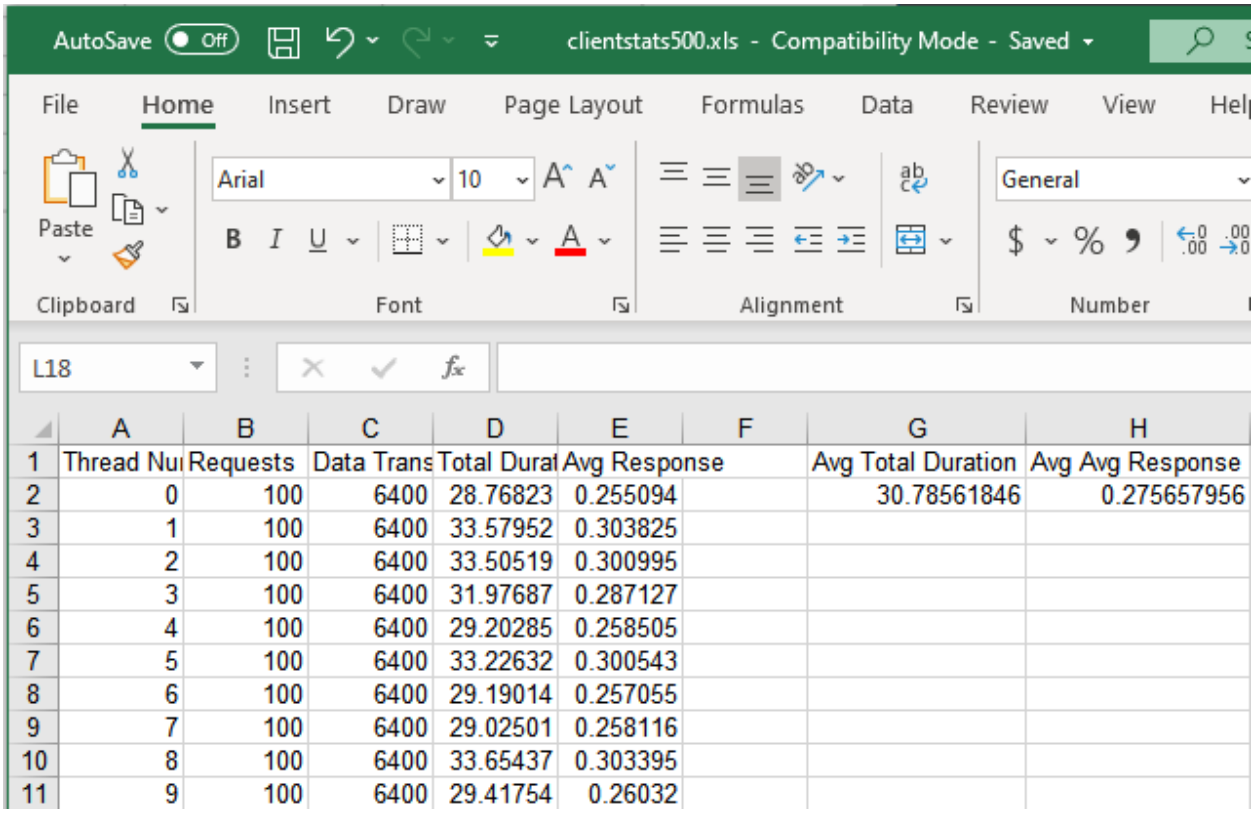


Server



Forwarder





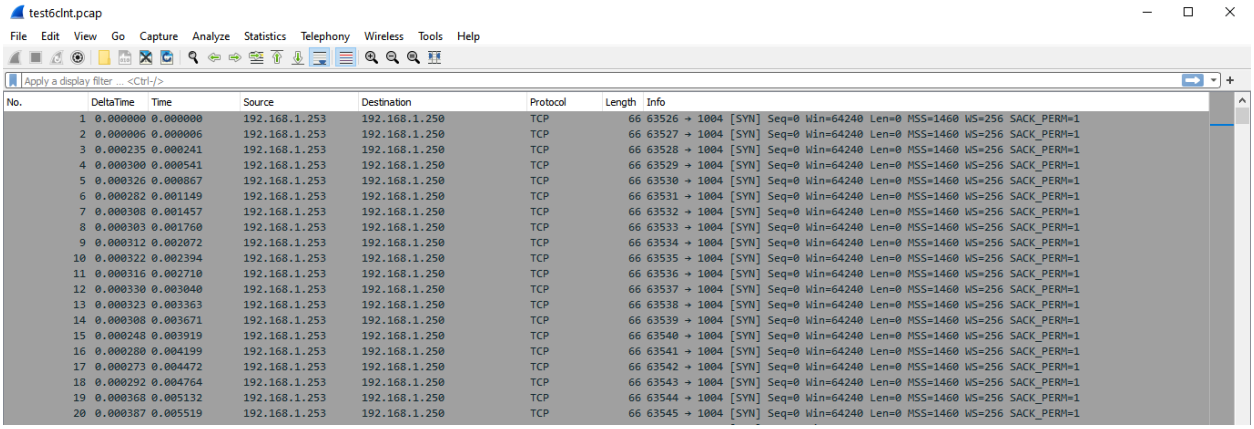
	A	B	C	D	E	F	G	H
1	Thread Nu	Requests	Data Trans	Total Dural	Avg Response		Avg Total Duration	Avg Avg Response
2	0	100	6400	28.76823	0.255094		30.78561846	0.275657956
3	1	100	6400	33.57952	0.303825			
4	2	100	6400	33.50519	0.300995			
5	3	100	6400	31.97687	0.287127			
6	4	100	6400	29.20285	0.258505			
7	5	100	6400	33.22632	0.300543			
8	6	100	6400	29.19014	0.257055			
9	7	100	6400	29.02501	0.258116			
10	8	100	6400	33.65437	0.303395			
11	9	100	6400	29.41754	0.26032			

The average total duration and average response time has increased by almost 3x when compared to having no forwarder

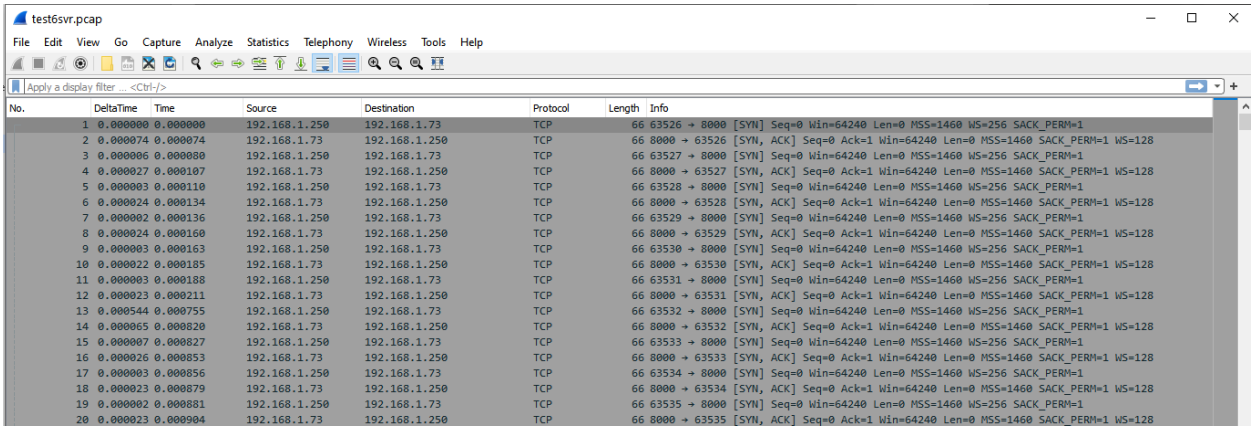
7.6 Test 6 1000 Clients

The performance of the port forwarder with 1000 clients coming from 192.168.1.253 is very slow and successfully completes the test

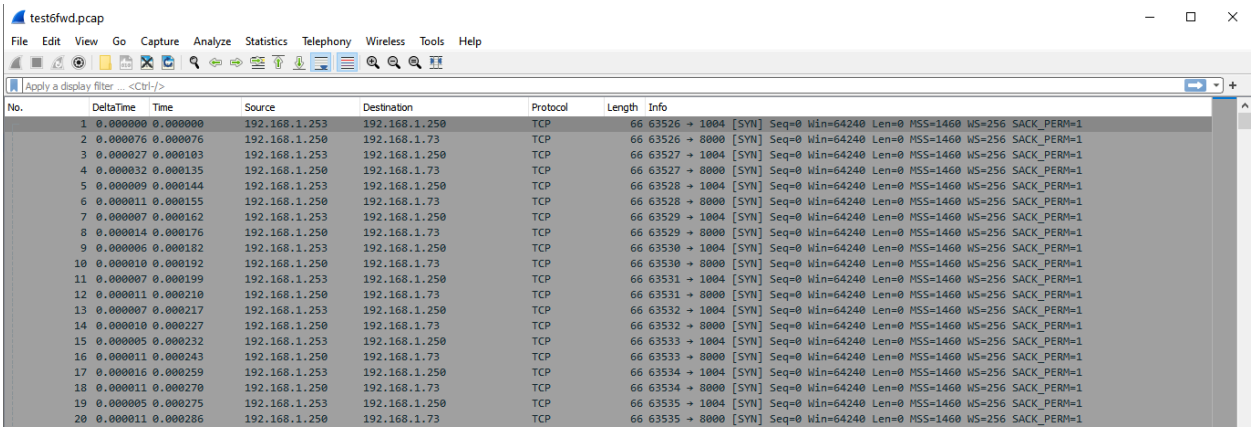
Client

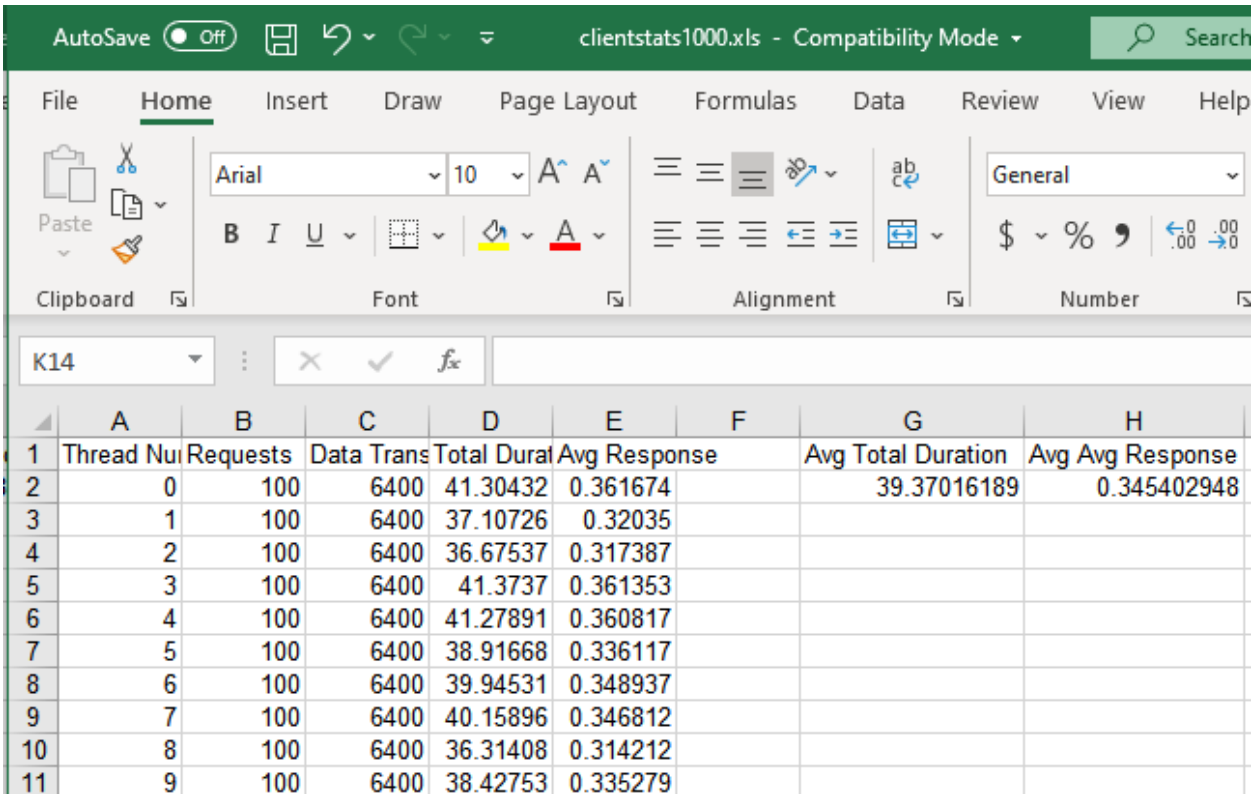


Server



Forwarder





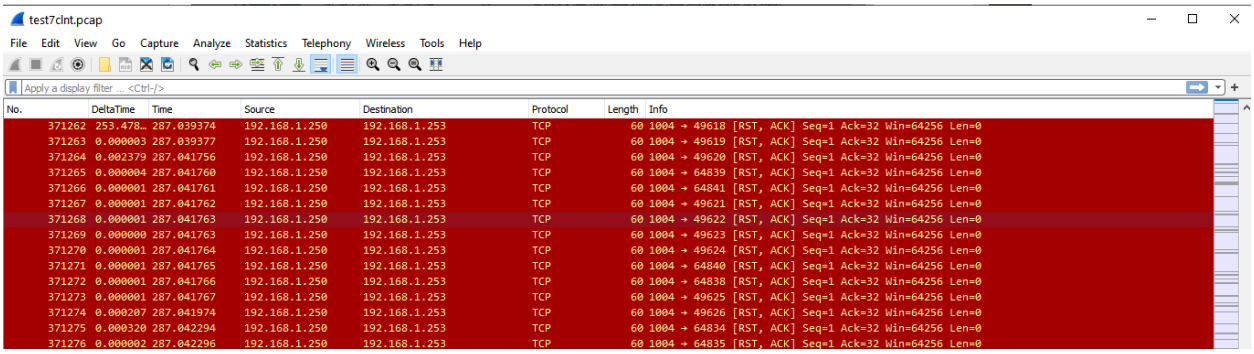
	A	B	C	D	E	F	G	H
1	Thread Nu	Requests	Data Trans	Total Durat	Avg Response		Avg Total Duration	Avg Avg Response
2	0	100	6400	41.30432	0.361674		39.37016189	0.345402948
3	1	100	6400	37.10726	0.32035			
4	2	100	6400	36.67537	0.317387			
5	3	100	6400	41.3737	0.361353			
6	4	100	6400	41.27891	0.360817			
7	5	100	6400	38.91668	0.336117			
8	6	100	6400	39.94531	0.348937			
9	7	100	6400	40.15896	0.346812			
10	8	100	6400	36.31408	0.314212			
11	9	100	6400	38.42753	0.335279			

The average total duration and average response time has increased by almost 2.3x when compared to having no forwarder

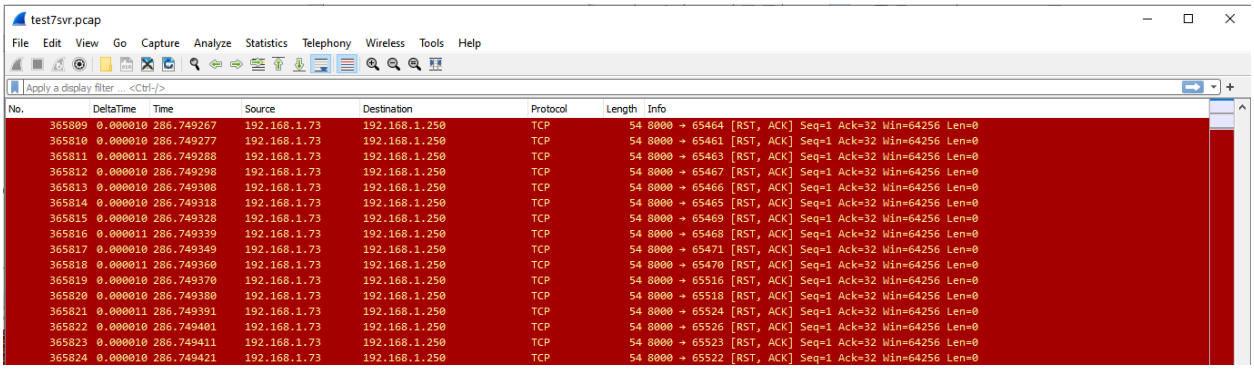
7.7 Test 7 1500 Clients

The performance of the port forwarder with 1500 clients coming from 192.168.1.253 locks up for multiple minutes before I decided to end and say the test has failed

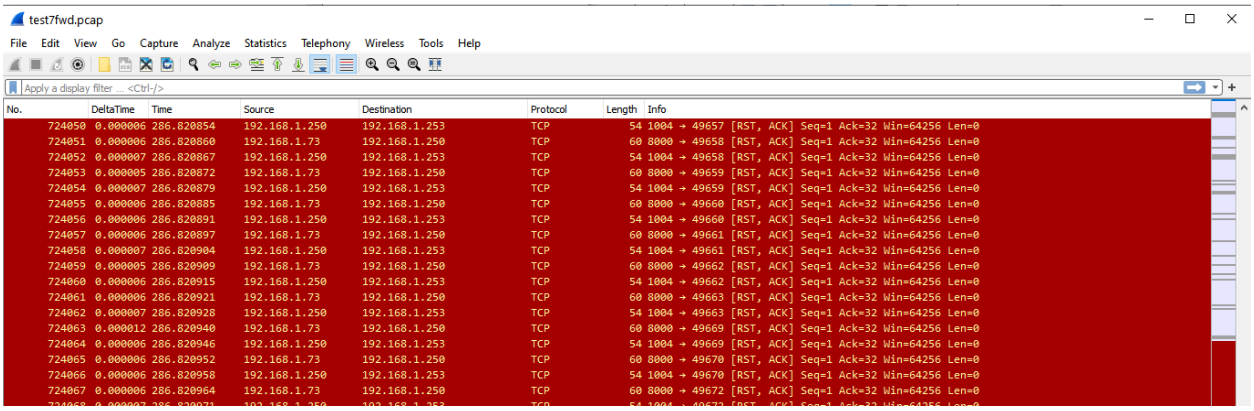
Client



Server



Forwarder

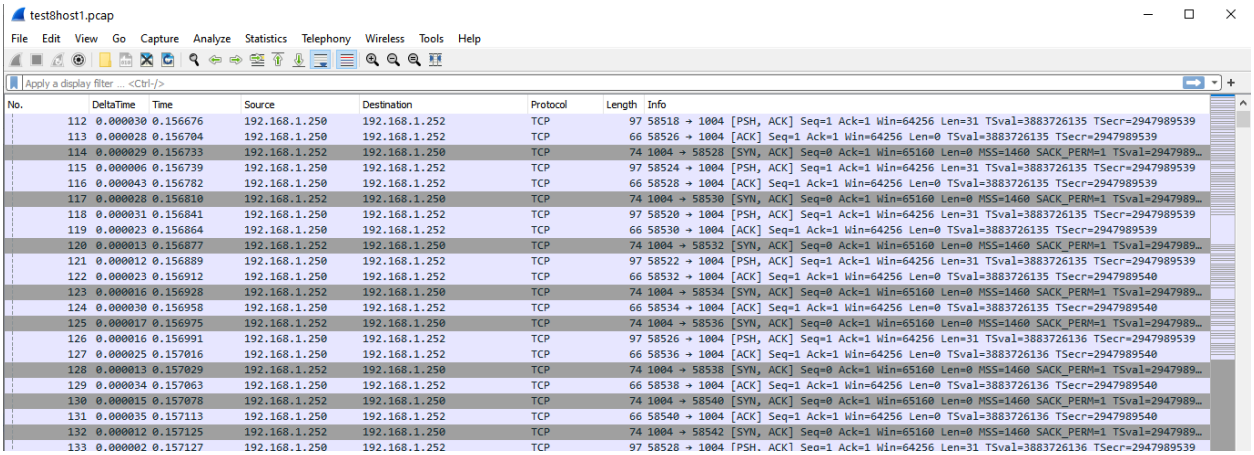


Because the test failed, I am unable to retrieve the excel spreadsheet of the client response times, I only have the server log which shows that about 500 connections did not finish.

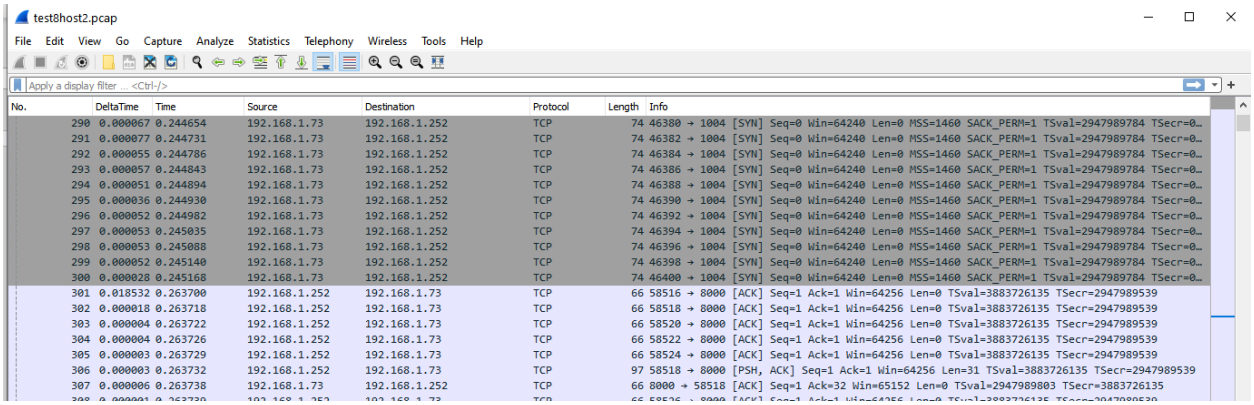
7.8 Test 8 Simultaneous Two-way Traffic 100+100 Clients

The test of simultaneous two-way traffic between two computers by running both epoll server and echo server on both hosts and having them port forward their packets using the RPI4 as the port forwarder 192.168.1.252 worked successfully.

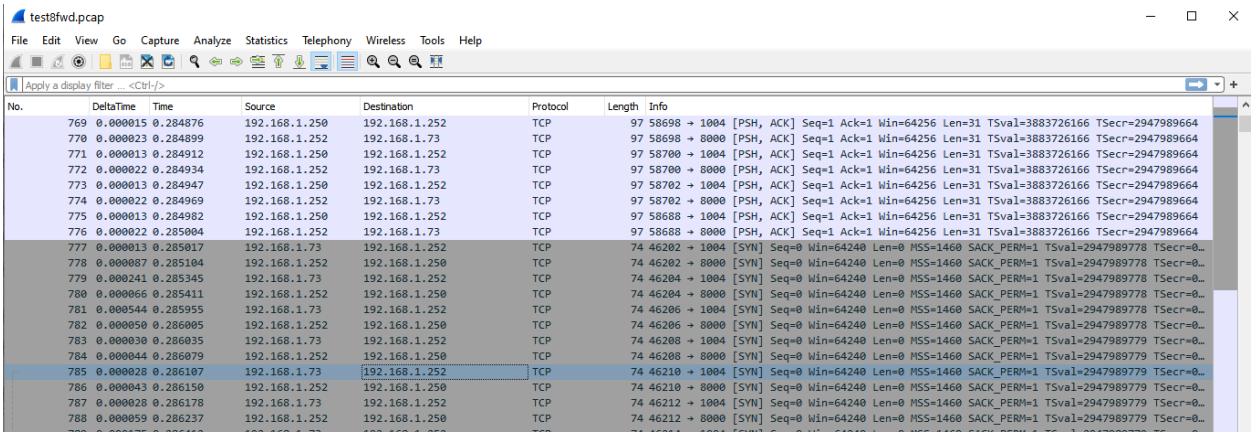
Host1



Host2



Forwarder



8 Conclusions

Using iptables to do port forwarding is very simplistic and I would very highly recommend it as it is only a few lines of code to get working. The speed difference between having a port forwarder and not having one between the client and epoll server is massive with about a speed penalty on both response and connection durations of about 3x.

8.1 Table Comparison between Forwarded and Non Forwarded

Connections	Status	Requests	Total Requests	AvgTotalDuration	AvgResponseTime
100	Success	100	10000	9.089017622	0.055820329
500	Success	100	50000	30.78561846	0.275657956
1000	Success	100	100000	39.37016189	0.345402948
1500	Fail				

Figure 1 Port Forwarded

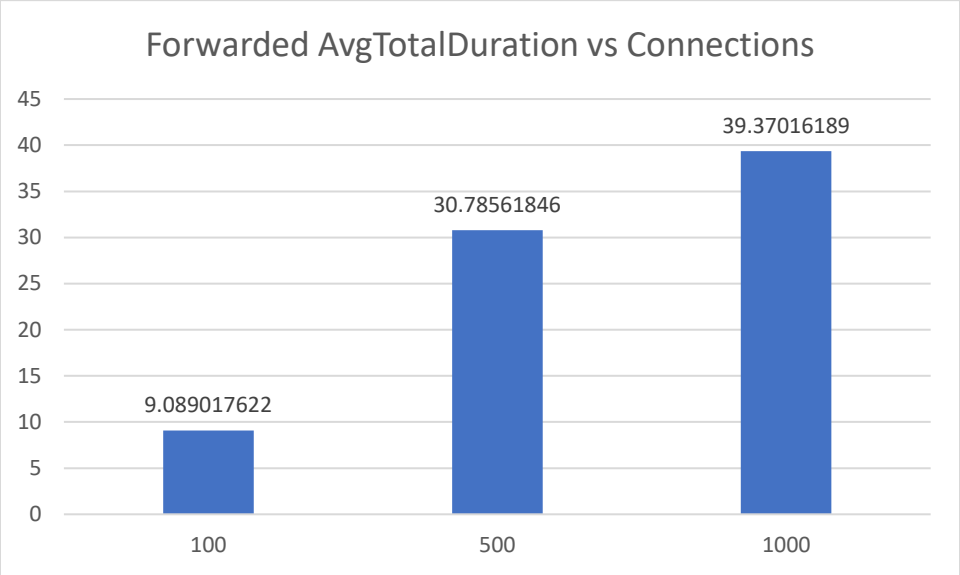
Connections	Status	Requests	Data Transferred(Bytes)	Total Requests	Total Data Transferred(Bytes)	AvgTotalDuration	AvgResponseTime
100	Success	100	6400	10000	64000000	3.559877663	0.016237825
200	Success	100	6400	20000	128000000	4.786482978	0.028205156
500	Success	100	6400	50000	320000000	9.951934278	0.079521272
1000	Success	100	6400	100000	640000000	17.88672988	0.155839739
1500	Success	100	6400	150000	960000000	25.72535474	0.231552899
2000	Success	100	6400	200000	1280000000	31.70220095	0.298103087
2500	Success	100	6400	250000	1600000000	37.14167742	0.354454643
3000	Success	100	6400	300000	1920000000	42.67918426	0.406414569
4000	Fail						

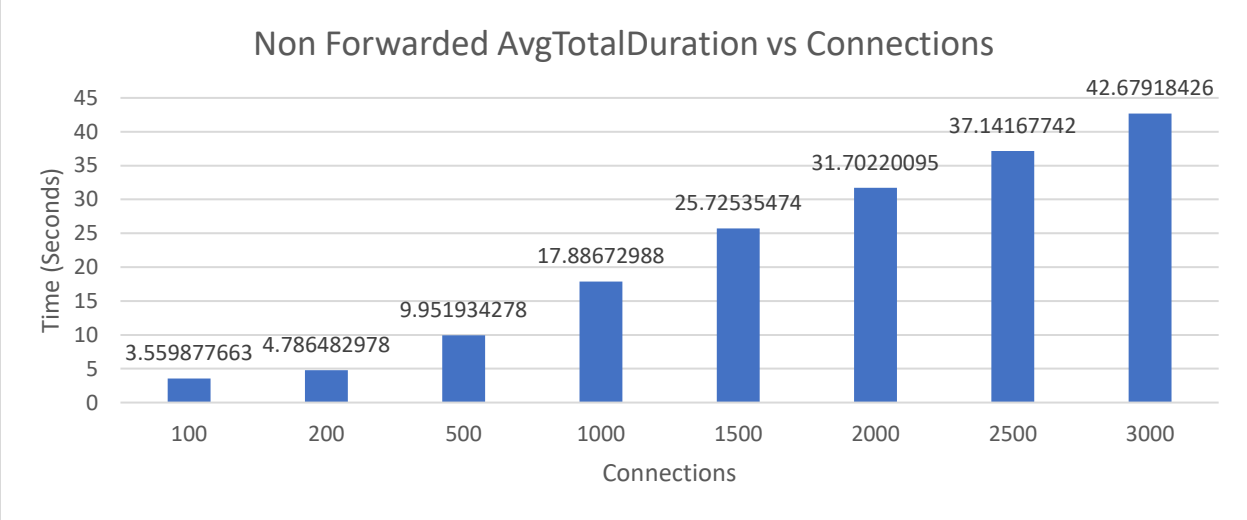
Figure 2 Non Port Forwarded

As shown, the speed differences are huge, 3x is a massive loss. Although its only a few milliseconds when theres few connections, it easily adds up when increasing the number of connections.

8.2 Bar Graph Comparison

8.2.1 AvgTotalDuration





8.2.2 AvgResponseTime