CS 204, Spring 2017
Abhishek Kumar SRIVASTAVA
Student ID: 861307778

Multipath-TCP Assignment
May 15, 2017

Solution . Setup Screenshots:

```
enx70886b801f3c: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.108  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::c4dc:31c9:1936:6f3d  prefixlen 64  scopeid 0x20<link>
        ether 70:88:6b:80:1f:3c  txqueuelen 1000  (Ethernet)
        RX packets 7155  bytes 5805501 (5.5 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7076  bytes 805748 (786.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 1: Ethernet Interface.

```
wlp2s0b1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.43.48  netmask 255.255.255.0  broadcast 192.168.43.25
5
        inet6 fe80::99f2:fef3:1d86:41ef  prefixlen 64  scopeid 0x20<link>
        ether e4:d5:3d:7b:9c:fe  txqueuelen 1000  (Ethernet)
        RX packets 858  bytes 128452 (125.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1374  bytes 165863 (161.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 2: Wifi Interface 1.

```
wlp2s0b1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.57  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::6eb9:a1e:c349:345f  prefixlen 64  scopeid 0x20<link>
        ether e4:d5:3d:7b:9c:fe  txqueuelen 1000  (Ethernet)
        RX packets 445  bytes 108885 (106.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 400  bytes 58296 (56.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 3: Wifi Interface 2.

```
$ sudo ip rule add from 192.168.1.108 table 1
$ sudo ip rule add from 192.168.43.48 table 2
$
$
$ sudo ip route add 192.168.1.0/24 dev enx70886b801f3c scope link table 1
$ sudo ip route add default via 192.168.1.1 dev enx70886b801f3c table 1
$
$
$ sudo ip route add 192.168.43.0/24 dev wlp2s0b1 scope link table 2
$ sudo ip route add default via 192.168.43.1 dev wlp2s0b1 table 2
$
$
$
$ sudo ip route add default scope global nexthop via 192.168.1.1 dev enx70
886b801f3c
$
```

Figure 4: Command to add route table.

```
$
$
$ ip rule show
0:      from all lookup local
32763:  from 192.168.43.48 lookup 2
32764:  from 192.168.1.57 lookup 3
32765:  from 192.168.1.108 lookup 1
32766:  from all lookup main
32767:  from all lookup default
$
$
$
```

Figure 5: IP Rule Show.

```
$
$
$ ip route
default via 192.168.1.1 dev enx70886b801f3c
default via 192.168.1.1 dev enx70886b801f3c  proto static  metric 100
default via 192.168.43.1 dev wlp2s0b1  proto static  metric 600
169.254.0.0/16 dev enx70886b801f3c  scope link  metric 1000
192.168.1.0/24 dev enx70886b801f3c  proto kernel  scope link  src 192.168.
1.108
192.168.1.0/24 dev enx70886b801f3c  proto kernel  scope link  src 192.168.
1.108  metric 100
192.168.43.0/24 dev wlp2s0b1  proto kernel  scope link  src 192.168.43.48
 metric 600
$
```

Figure 6: Routing Tables.

```
$
$
$ ip route show table 1
default via 192.168.1.1 dev enx70886b801f3c
192.168.1.0/24 dev enx70886b801f3c  scope link
$
$
$
$ ip route show table 2
default via 192.168.43.1 dev wlp2s0b1
192.168.43.0/24 dev wlp2s0b1  scope link
$
$
$
```

Figure 7: IP Route for table 1 & 2.

```
$
$
$ dmesg | grep MPTCP
[    0.848382] MPTCP: Stable release v0.91.3
$
$
$ sysctl -w net.mptcp.mptcp_enabled=1
sysctl: permission denied on key 'net.mptcp.mptcp_enabled'
$ sudo sysctl -w net.mptcp.mptcp_enabled=1
net.mptcp.mptcp_enabled = 1
$ sudo sysctl -w net.mptcp.mptcp_path_manager=1
sysctl: setting key "net.mptcp.mptcp_path_manager": No such file or direct
ory
net.mptcp.mptcp_path_manager = 1
$ sudo sysctl  net.mptcp.mptcp_path_manager
net.mptcp.mptcp_path_manager = fullmesh
$
$
$
```

Figure 8: Enabling MPTCP and Path Manager.

```
$
$
$
$ curl http://www.multipath-tcp.org
Yay, you are MPTCP-capable! You can now rest in peace.
$
$
```

Figure 9: MPTCP Enabling Testing.

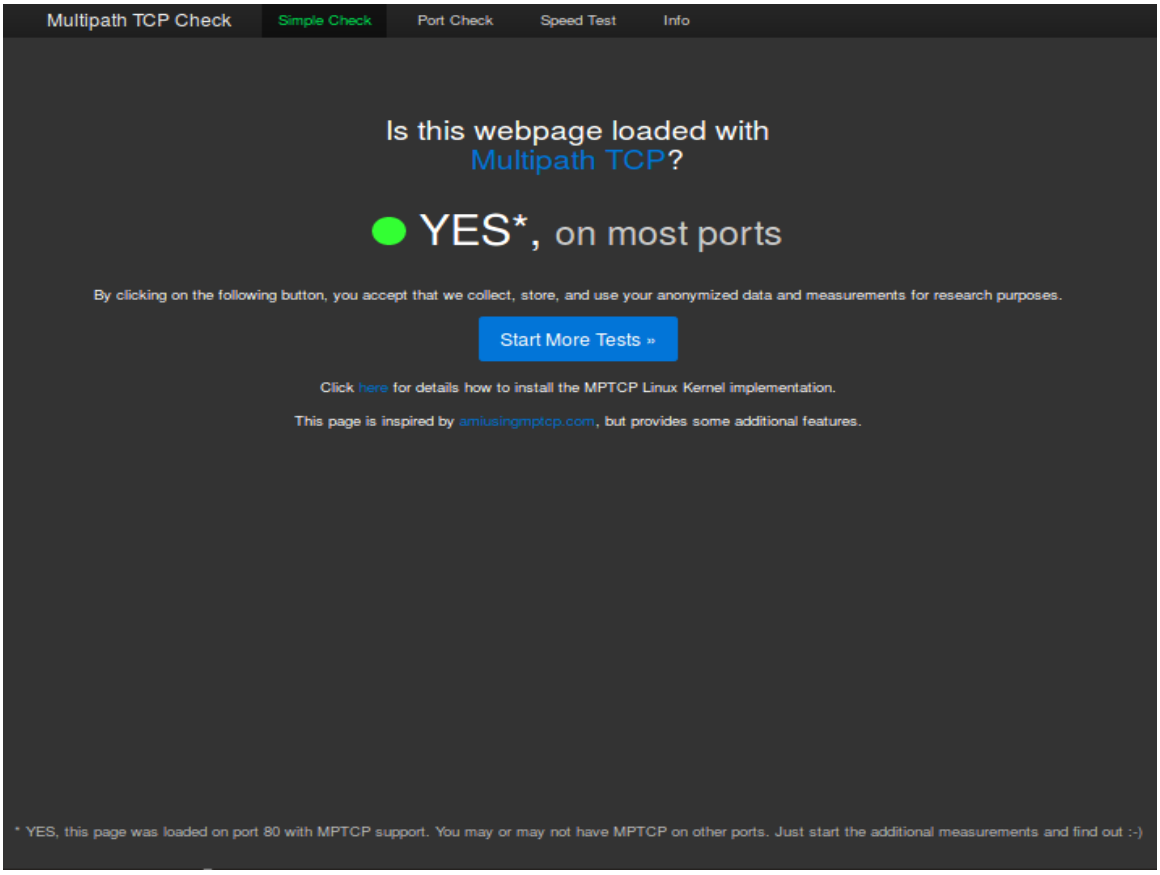

Figure 10: MPTCP Enabling Testing.

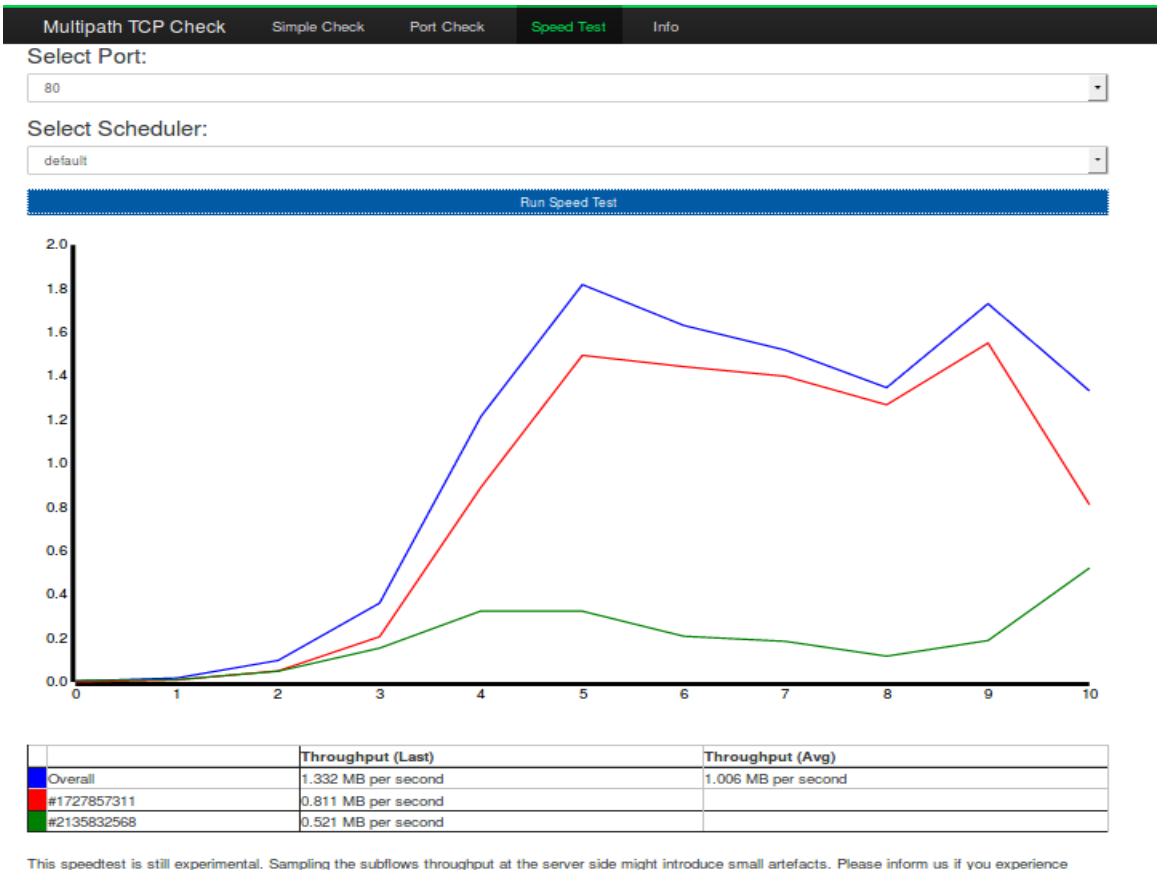

Figure 11: MPTCP Speed Test:1

Figure 12: MPTCP Speed Test:2

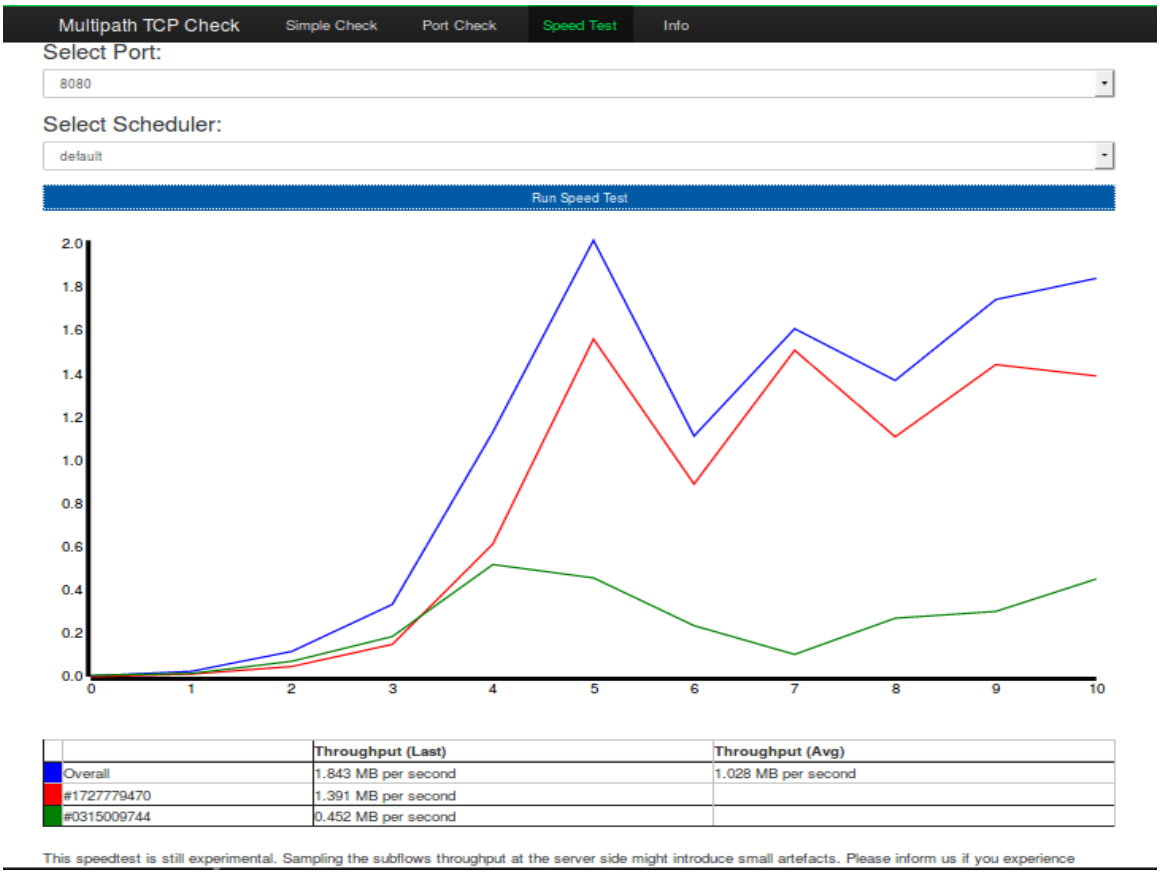

Figure 13: MPTCP Speed Test:3

```
$ sudo tcpdump -i any -w mptcp_001.pcap
[sudo] password for abhishek:
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked), capture siz
e 262144 bytes
^C112040 packets captured
112056 packets received by filter
0 packets dropped by kernel
$
$
$
```

Figure 14: Capturing pcap using tcpdump for mptcp interfaces.

```
$
$
$ sudo tcpdump -i any -w wifi_001.pcap
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked), capture siz
e 262144 bytes
^C54694 packets captured
54724 packets received by filter
0 packets dropped by kernel
$
$
$
```

Figure 15: Capturing pcap using tcpdump for wifi interface.

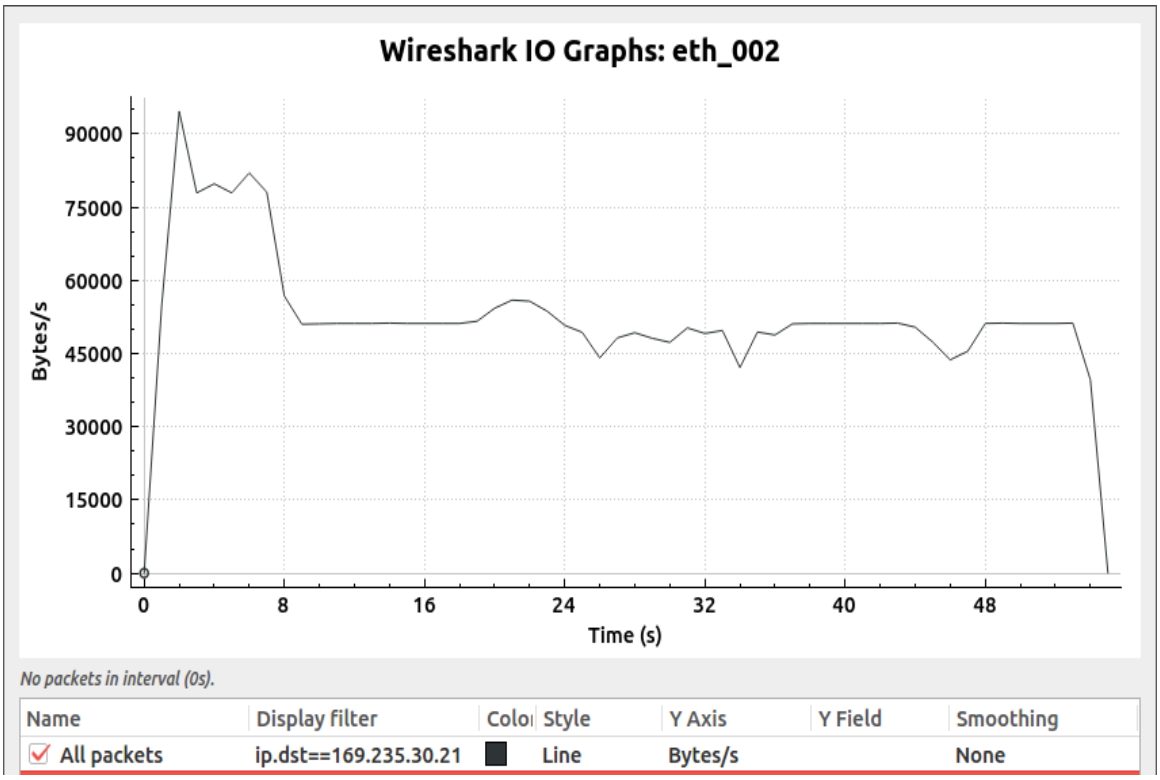

Figure 16: Through put using Ethernet interface only.

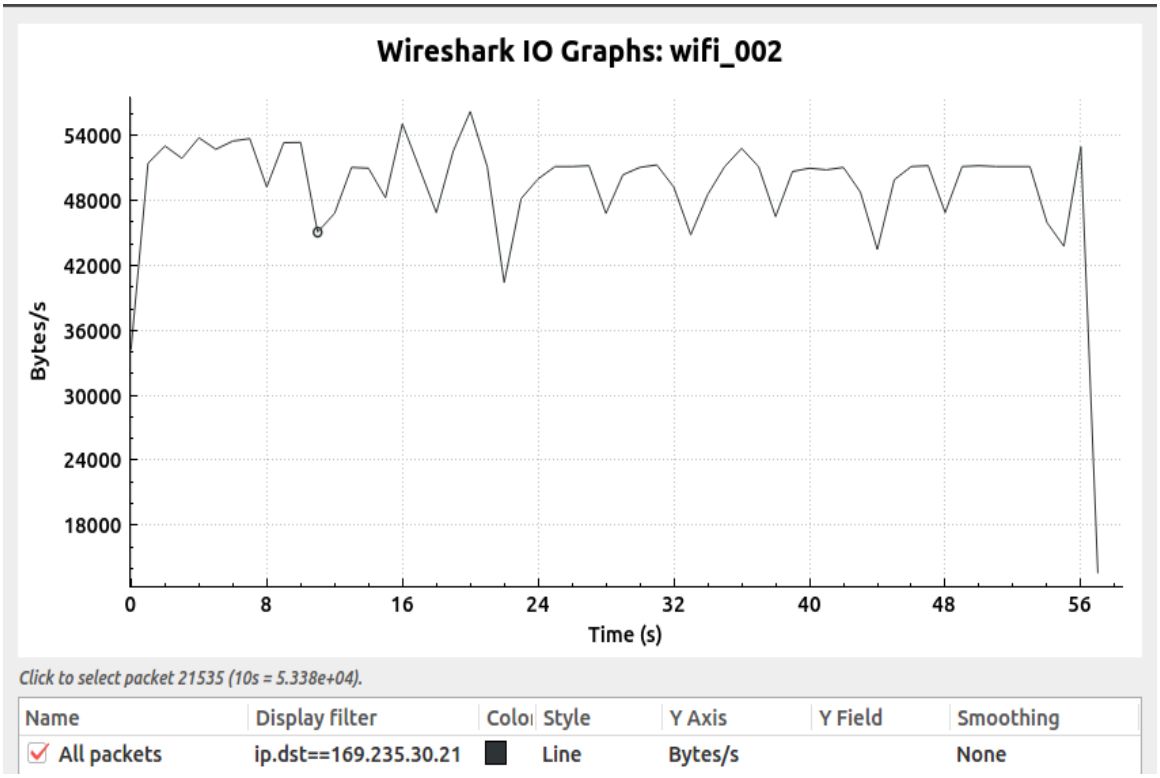

Figure 17: Through put using Wifi interface only.

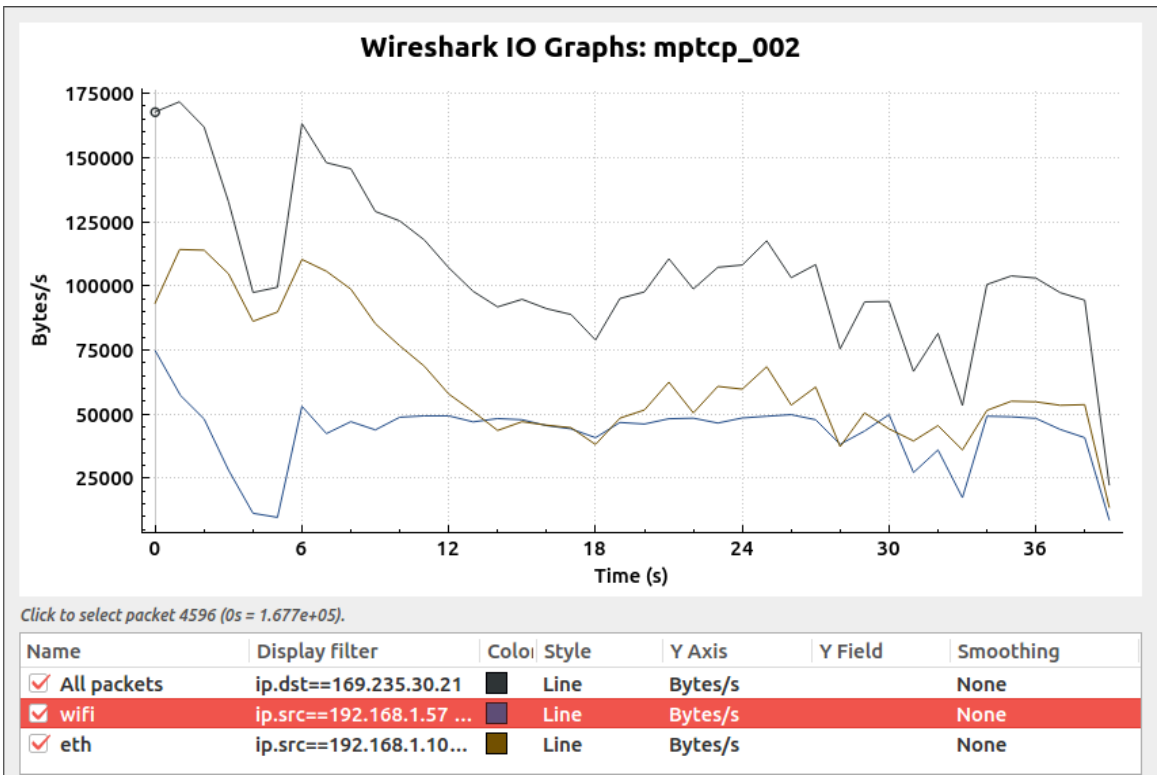

Figure 18: Through put using MPTCP interface

# Website	pcap	DOMContentLoaded	Load	Obj requested
http://mptcp.cs.ucr.edu/cnn.html	Ethernet			
	cnn_0e	4.01s	4.90s	86
	cnn_1e	3.26s	3.95s	3
	cnn_2e	2.87s	3.45s	4
	cnn_3e	2.78s	3.38s	4
	cnn_4e	4.34s	5.10s	86
	WiFi			
	cnn_0w	4.37s	5.38s	86
	cnn_1w	2.89s	3.52s	4
	cnn_2w	2.81s	3.34s	4
	cnn_3w	2.81s	3.47s	4
	cnn_4w	3.68s	5.70s	85
	MPTCP			
	cnn_0m	3.57s	4.66s	86
	cnn_1m	2.72s	3.39s	4
	cnn_2m	2.77s	3.30s	4
	cnn_3m	2.94s	3.51s	4
	cnn_4m	3.70s	4.44s	86

Figure 19: Table for CNN Website load for Ethernet, Wifi and MPTCP interface.

# Website	pcap	DOMContentLoaded	Load	Obj requested
http://mptcp.cs.ucr.edu/foxnews.html	Ethernet			
	fox_0e	2.05s	4.23s	69
	fox_1e	1.82s	4.09s	12
	fox_2e	1.91s	4.96s	13
	fox_3e	1.79s	4.22s	13
	fox_4e	2.10s	5.04s	69
	WiFi			
	fox_0w	2.28s	4.61s	68
	fox_1w	1.95s	4.82s	14
	fox_2w	1.92s	4.92s	11
	fox_3w	2.01s	5.02s	12
	fox_4w	2.67s	4.72s	68
	MPTCP			
	fox_0m	1.93s	4.11s	68
	fox_1m	1.98s	4.81s	13
	fox_2m	1.80s	4.05s	10
	fox_3m	1.99s	4.11s	12
	fox_4m	2.12s	4.22s	69

Figure 20: Table for Fox Website load for Ethernet, Wifi and MPTCP interface.

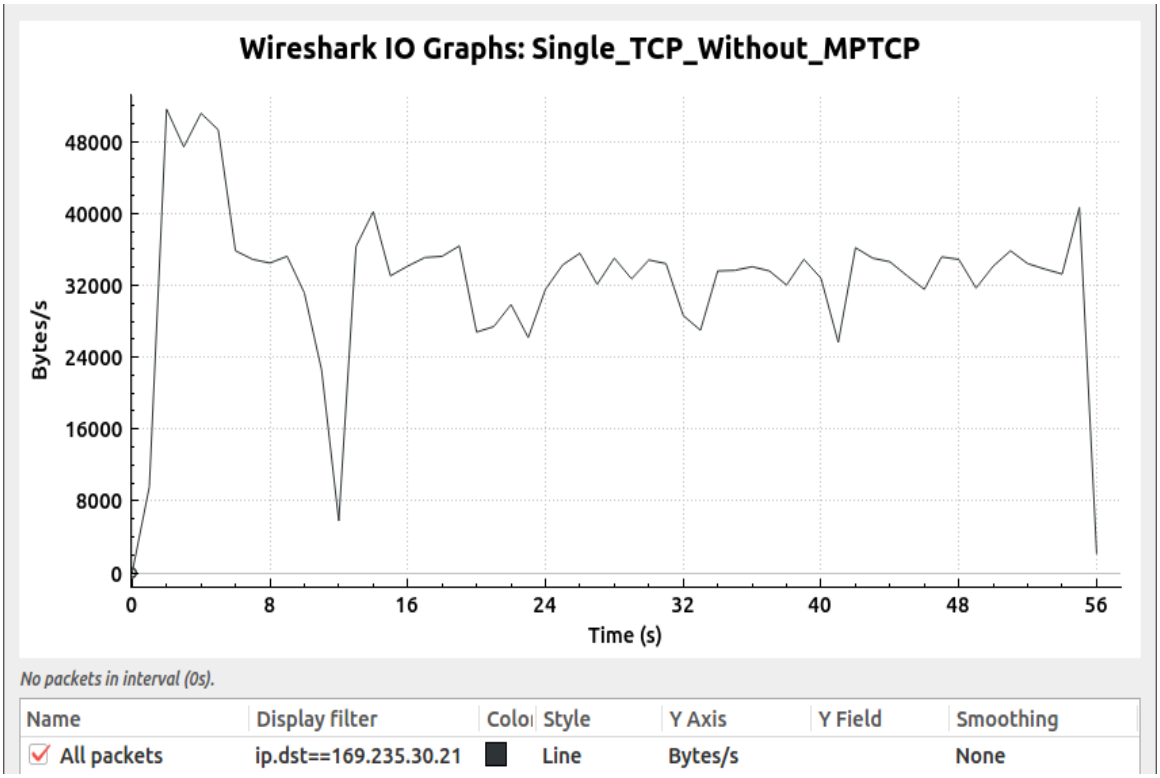

Figure 21: Throughput of Single TCP without MPTCP connection.

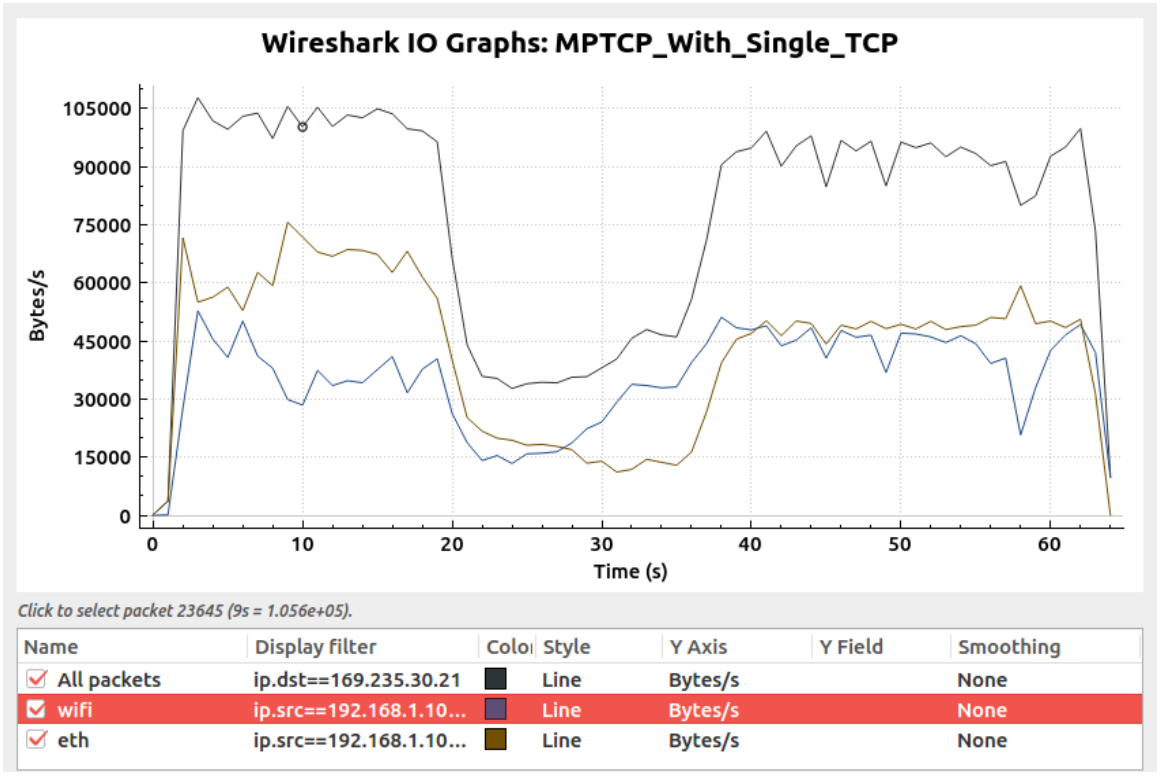

Figure 22: Throughput of MPTCP with Single TCP connection.

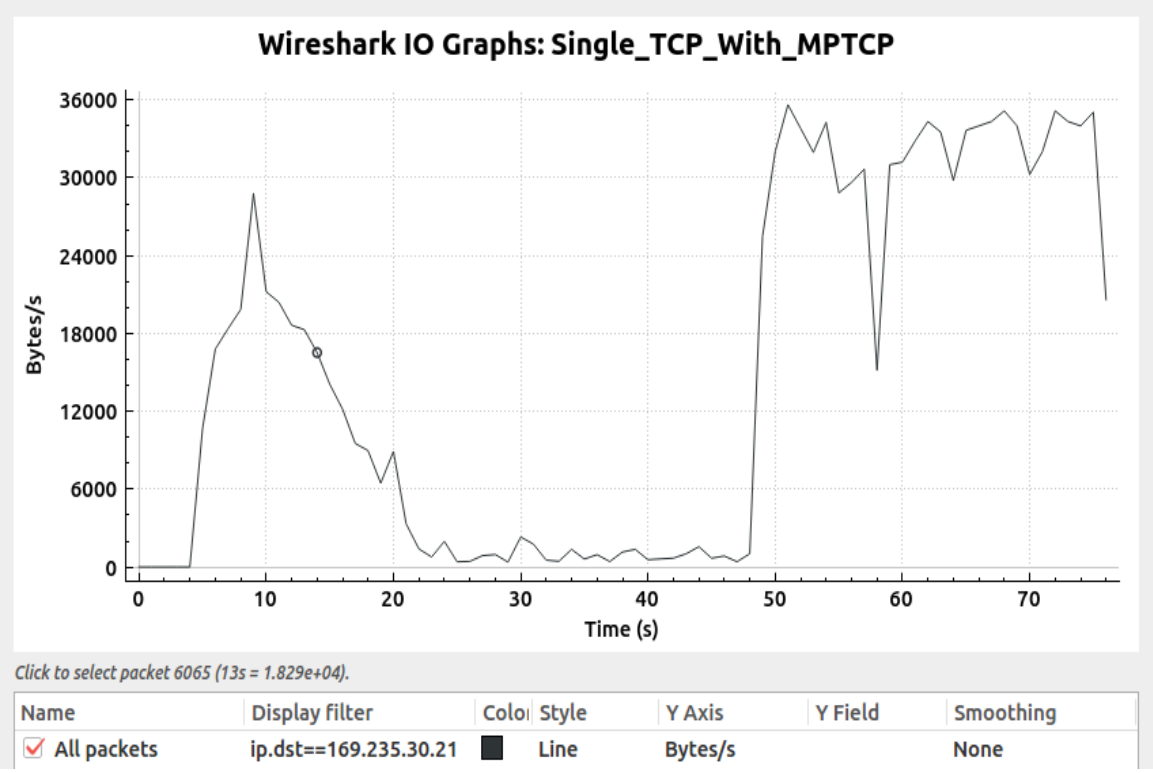

Figure 23: Throughput of single TCP in presence of MPTCP connection.

```
Q1. Does MPTCP improve file transfer performance?


Questions: Does MPTCP increase the total throughput? How does the traffic split using MPTCP compare to the
single-path scenarios?

Yes, As from Figure 16,17 and 18 we can see that using only ethernet we could only send about
60000 Bytes/s on avergave and 48000 Bytes/s using Wifi interface on average but when we try downloading
it using MPTCP we can see that it was able to send 100000 Bytes/s. Which does imply that MPTCP did
improved the file transfer performance.

MPTCP did increase the total throughput. Figure 18 shows the distribution of traffic split between Wifi
and Ethernet interfaces. Intially Ethernet was quite dominant in traffic but later on Wifi and Ethernet
contirbuted same amount of traffic/ throughput.
```

Figure 24: Answer 1.

2. Does MPTCP improve web browsing performance?

Questions: Does MPTCP help reduce the page load time? Why or why not? How does the fraction of objects requested correlate to the throughput from the previous step?

Yes, MPTCP does improve web browsing performance by some margin, it is specially fast when whole page is loaded out without using cached data. In my testing Figure 19 and 20, the first reading is while loading whole page after clearing the cache, then 3 readings are loading the page using cached items and then last reading without cached items as well. We can clearly see the difference in terms of objectes fetched.

To some extent MPTCP does decrease the page load time, decrease time in loading is more prevalent when webpage is fetched without using chached items. The reason for being decreased load time is becuase faster fetching of the objects. If more objects are to be fetched and if MPTCP path is used it will fecth all the objects at faster rate, hence decrease in the load time.

Figure 25: Answer 2.

3. Is MPTCP TCP-friendly?

Questions: How does the MPTCP client's throughput change when the single-path user joins? Comparing their throughputs, is the MPTCP client TCP-friendly? How does the other interface of the MPTCP client adapt when the single-path client joins?

From the results obtained I did not find out MPTCP to be TCP-friedly.
Figure 21 shows Single TCP throughput without any MPTCP connection over the same interface.
Figure 22 shows throughput of MPTCP with single path TCP connection as well. As you can see after about 20 seconds, the throughput decreases for the MPTCP is because Single path TCP connection is established and total thorughput is being shared between MPTCP and TCP. But after few seconds we can see that MPTCP again starts dominating. In figure 23 we can see Single path TCP throughput in presence of MPTCP connection, intially it started with good throughput but in some time it falls off and keep decreasing and once the MPCTP connection is closed, It agains starts to gain more and more throughput.

Figure 26: Answer 3.