Assignment 2 – data testing

Ping tests:

1. lab → pi, 10 times 0.2s interval
--- 192.168.10.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 1836ms
rtt min/avg/max/mdev = 0.433/0.476/0.511/0.019 ms

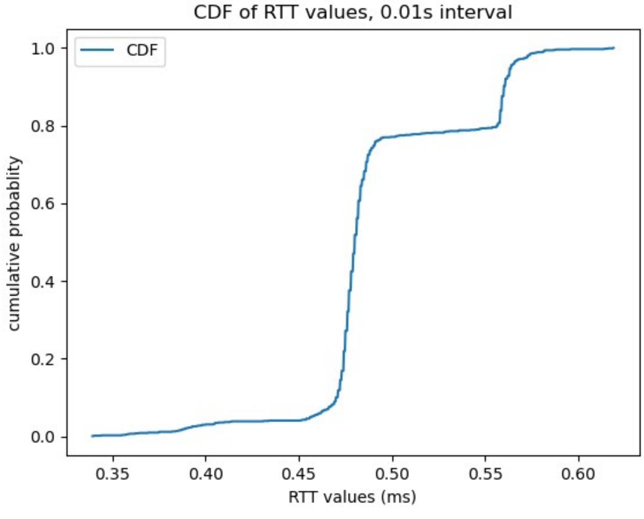
2. pi → lab, same conditions as above
--- 192.168.10.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 1841ms
rtt min/avg/max/mdev = 0.402/0.530/0.646/0.072 ms

3. pi → lab, 100 times, 0.001s interval
--- 192.168.10.1 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99ms
rtt min/avg/max/mdev = 0.308/0.348/0.550/0.031 ms

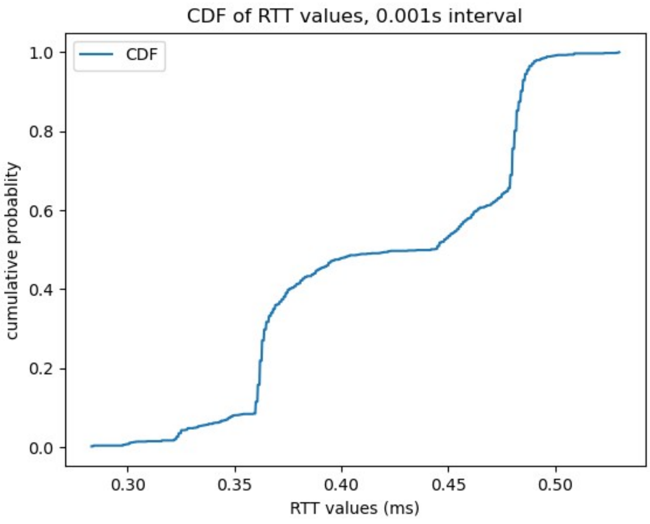
the mean is much closer to the minimum than the maximum. This is different to the first two tests where the mean was close to between the min and max.
This is likely due to the fact that we are running more tests so there is a higher likelihood that a ping is sent during a time of high congestion, high packet loss or when other high priority signals are being sent. This will mean our signal will have to wait a bit before being sent therefore increasing the ping in some cases as can be seen in the max case.

4. pi → lab, 10000 times, with flooding
--- 192.168.10.1 ping statistics ---
10000 packets transmitted, 10000 received, 0% packet loss, time 3337ms
rtt min/avg/max/mdev = 0.161/0.299/0.495/0.024 ms, ipg/ewma 0.333/0.297 ms

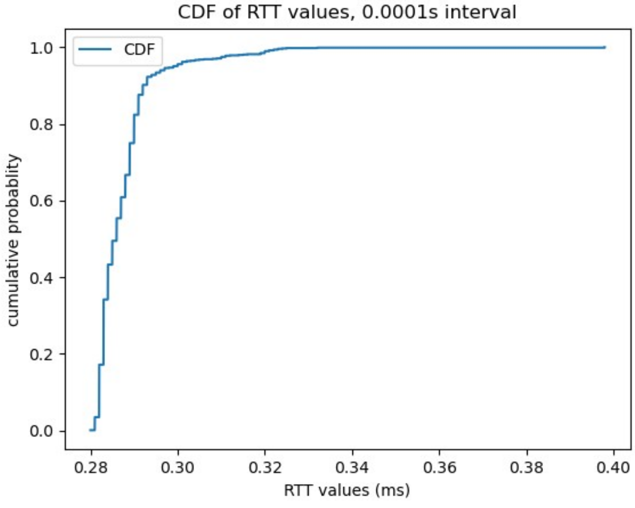
5.
CDF for 0.01s interval



0.001s interval



0.0001s interval



6.

Ping provides an upper-bound value for the RTT as it also takes into consideration, and measures the time taken to get a path for our ping message to be relayed through. This time can be quite significant due to factors such as network congestion, prioritization, latency and packet loss. When running at higher time intervals, there is a greater chance of encountering these situations which will increase the ping times therefore increasing the max and mean times and skewing the CDF to the right.

This means that the most accurate RTT would probably be the minimum time.

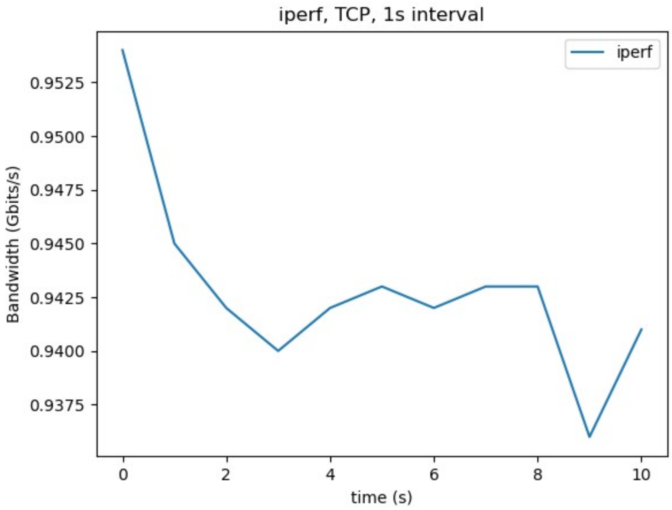
Iperf Tests:

1. lab machine as server, pi as client. 10 sec experiment
[ID] Interval   Transfer   Bandwidth
[3] 0.0000-10.0030 sec 1.10 GBytes 0.941 Gbits/sec

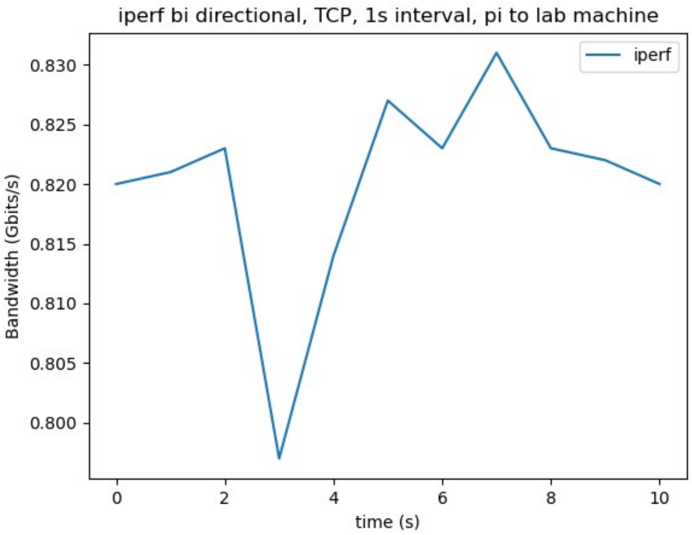
2.

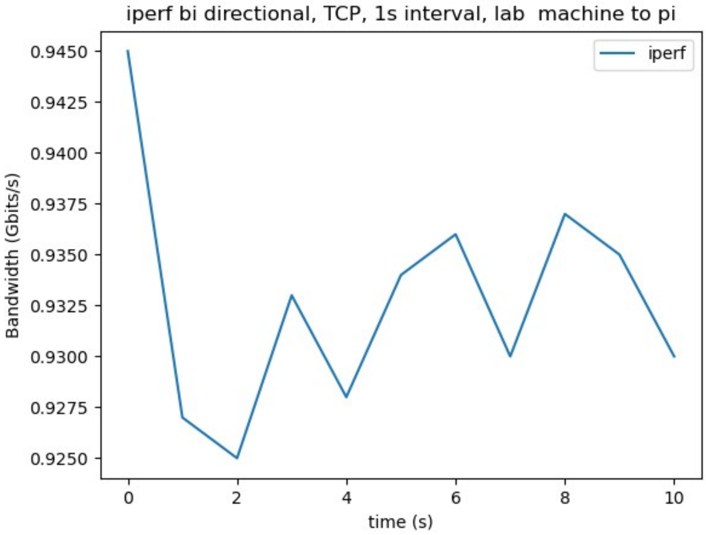
[ID] Interval   Transfer   Bandwidth
[1] 0.0000-1.0000 sec 0.111 GBytes 0.954 Gbits/sec
[1] 1.0000-2.0000 sec 0.110 GBytes 0.944 Gbits/sec
[1] 2.0000-3.0000 sec 0.110 GBytes 0.942 Gbits/sec
[1] 3.0000-4.0000 sec 0.110 GBytes 0.942 Gbits/sec
[1] 4.0000-5.0000 sec 0.109 GBytes 0.937 Gbits/sec
[1] 5.0000-6.0000 sec 0.110 GBytes 0.942 Gbits/sec
[1] 6.0000-7.0000 sec 0.110 GBytes 0.942 Gbits/sec
[1] 7.0000-8.0000 sec 0.110 GBytes 0.942 Gbits/sec
[1] 8.0000-9.0000 sec 0.110 GBytes 0.943 Gbits/sec
[1] 9.0000-10.0000 sec 0.110 GBytes 0.945 Gbits/sec
[1] 0.0000-10.0324 sec 1.10 GBytes 0.940 Gbits/sec

Graph:



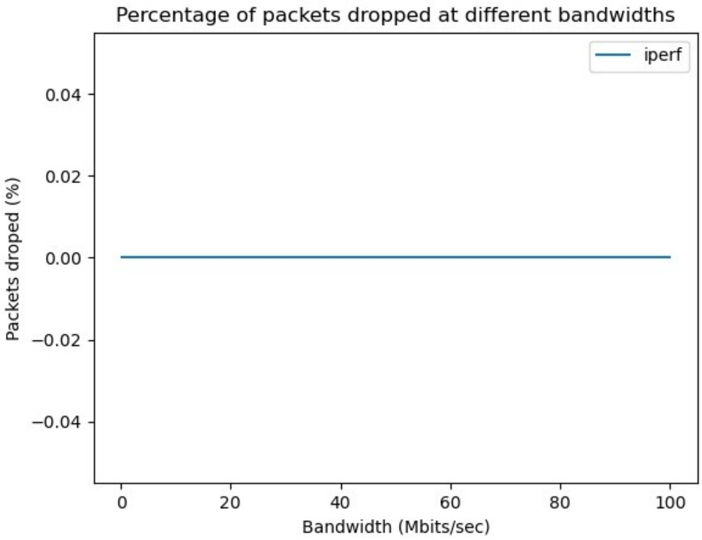
3.
pi as server, lab machine as client. Bi-directional test, 10s, 1s interval





4.

a. for all of the different bandwidths, no packets were dropped



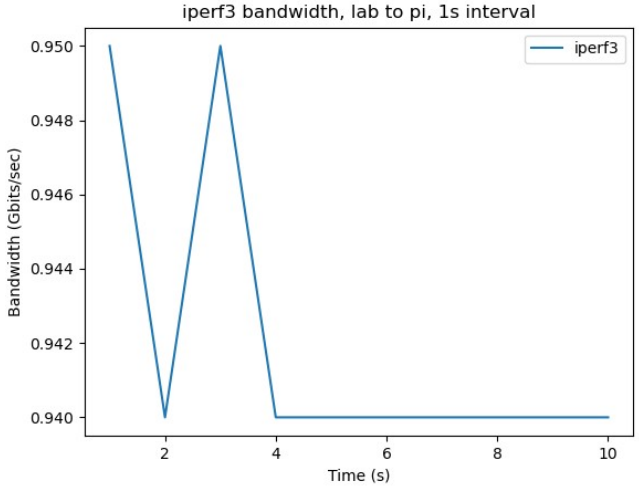
b.
100kb/s:
[ID] Interval   Transfer   Bandwidth
[3] 0.0000-1.0000 sec 13.0 KBytes   106 Kbits/sec
[3] 1.0000-2.0000 sec 12.9 KBytes   106 Kbits/sec
[3] 2.0000-3.0000 sec 11.5 KBytes 94.1 Kbits/sec
[3] 3.0000-4.0000 sec 12.9 KBytes   106 Kbits/sec
[3] 4.0000-5.0000 sec 11.5 KBytes 94.1 Kbits/sec

1Mb/s:
[ID] Interval   Transfer   Bandwidth
[3] 0.0000-1.0000 sec   124 KBytes 1.01 Mbits/sec
[3] 1.0000-2.0000 sec   122 KBytes 1000 Kbits/sec
[3] 2.0000-3.0000 sec   122 KBytes 1000 Kbits/sec
[3] 3.0000-4.0000 sec   122 KBytes 1000 Kbits/sec
[3] 4.0000-5.0000 sec   122 KBytes 1000 Kbits/sec

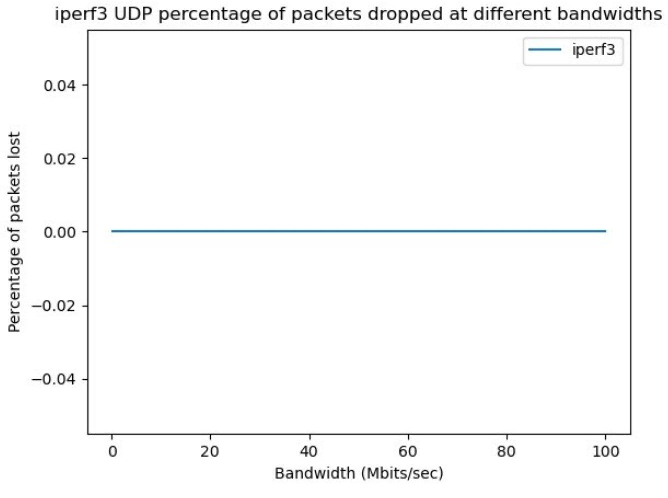
100Mb/s:

[ID] Interval   Transfer   Bandwidth
[3] 0.0000-1.0000 sec 11.9 MBytes   100 Mbits/sec
[3] 1.0000-2.0000 sec 11.9 MBytes   100 Mbits/sec
[3] 2.0000-3.0000 sec 11.9 MBytes   100 Mbits/sec
[3] 3.0000-4.0000 sec 11.9 MBytes   100 Mbits/sec
[3] 4.0000-5.0000 sec 11.9 MBytes   100 Mbits/sec

Iperf3 Test:
1.



2.
a.



b.
100Kb/s:
[5]   0.00-1.00   sec 12.7 KBytes 0.10 Mbits/sec 9
[5]   1.00-2.00   sec 12.7 KBytes 0.10 Mbits/sec 9
[5]   2.00-3.00   sec 11.3 KBytes 0.09 Mbits/sec 8
[5]   3.00-4.00   sec 12.7 KBytes 0.10 Mbits/sec 9
[5]   4.00-5.00   sec 12.7 KBytes 0.10 Mbits/sec 9

1Mb/s:
[5]   0.00-1.00   sec   123 KBytes 1.01 Mbits/sec 87

```
[  5]   1.00-2.00   sec   122 KBytes  1.00 Mbits/sec  86
[  5]   2.00-3.00   sec   122 KBytes  1.00 Mbits/sec  86
[  5]   3.00-4.00   sec   123 KBytes  1.01 Mbits/sec  87
[  5]   4.00-5.00   sec   122 KBytes  1.00 Mbits/sec  86
```

100Mb/s

```
[  5]   0.00-1.00   sec  11.9 MBytes  99.9 Mbits/sec  8626
[  5]   1.00-2.00   sec  11.9 MBytes   100 Mbits/sec  8633
[  5]   2.00-3.00   sec  11.9 MBytes   100 Mbits/sec  8632
[  5]   3.00-4.00   sec  11.9 MBytes   100 Mbits/sec  8633
[  5]   4.00-5.00   sec  11.9 MBytes   100 Mbits/sec  8632
```

3.
TCP test: looking at the two graphs of lab machine to I for iperf and iperf3. We see that Iperf3 performs much better as it is able to settle onto a value after only a few seconds whereas Iperf fluctuates by quite a lot through the 10s test making it difficult to find the true bandwidth

packet loss: in both iperf and iperf3 there are no packets lost showing both handle the UDP protocol well and reliably.

Looking at the raw results both iperf and iperf3 perform similarly. There are however small differences, Iperf3 usually transfers slightly less bytes to perform the test when compared with Iperf, especially for lower bandwidths. As well as this Iperf3 is more precise than iperf at lower Bandwidths, the precision is the same at higher ones.

These small differences are likely to due efficiencies and fixes as iperf was updated and its algorithms and mechanisms were improved.