Open in Colab

TCP Throughput v.s. Parallel Connections

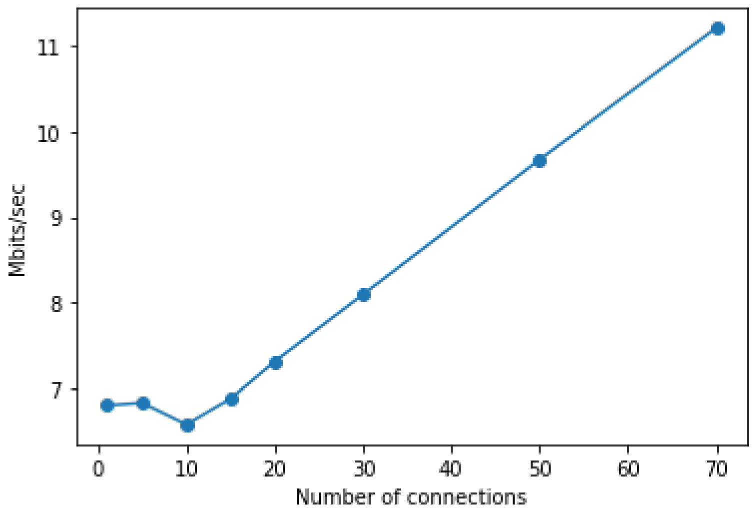
Script

```
#!/bin/sh
TIME=10
iperf3 -c bouygues.testdebit.info -p 5204 -t $TIME > p1.txt
iperf3 -c bouygues.testdebit.info -p 5204 -P 5 -t $TIME > p5.txt
iperf3 -c bouygues.testdebit.info -p 5204 -P 10 -t $TIME > p10.txt
iperf3 -c bouygues.testdebit.info -p 5204 -P 20 -t $TIME  > p20.txt
iperf3 -c bouygues.testdebit.info -p 5204 -P 30 -t $TIME  > p30.txt
iperf3 -c bouygues.testdebit.info -p 5204 -P 50 -t $TIME  > p50.txt
iperf3 -c bouygues.testdebit.info -p 5204 -P 70 -t $TIME  > p70.txt
iperf3 -c bouygues.testdebit.info -p 5204 -P 100 -t $TIME  > p100.txt
```

Plot

```python
import matplotlib.pyplot as plt
```

```python
speeds = [6.81, 6.84, 6.59, 6.89, 7.32, 8.10, 9.67, 11.2]
numConns = [1, 5, 10, 15, 20, 30, 50, 70]
plt.plot(numConns, speeds, 'o', ls='-')
plt.xlabel('Number of connections')
plt.ylabel('Mbits/sec')
plt.show()
```



Questions

1.

> Describe the parameters you use, e.g., length of time to transmit (with -t), number of parallel connections (with -P), and other parameters if you specify any. Note, only the number of parallel connections should vary, and all other parameters should be fixed.

I set the length of time to transmit to 10 seconds and ran the command with the numbers of connections=1, 5, 10, 15, 20, 30, 50, 70. (I tried `-P 100` but got Connection refused.)

2.

> What's the number of parallel connections that give the highest aggregate bandwidth? What's the trend of aggregate bandwidth as the number of parallel connections increases?

Number of parallel connections=70 gave the best result. The higher the number of connections, the higher bandwith I got.

3.

> Briefly explain the possible reasons behind what you observe. For instance, if aggregate bandwidth increases with more parallel connections at the beginning, then what's the mathematical formula that may explain this? And if aggregate bandwidth stops increasing or even decreases a bit later on, what might be the possible causes?

It makes sense that aggregate bandwith increases with more parallel connections resulted in higher bandwidth. As we can see in the plot above, the number of connections is in proportion to the bandwith. So an mathematical expression that describe the relationship would be $B=cN$, with B being the bandwith and N being the number of connections.
If the bandwith stops increasing at a point, it may be because the speed is bottlenecked by other factors such as hardware limitations.

TCP Throughput v.s. Round Trip Time (RTT) to Server

Script

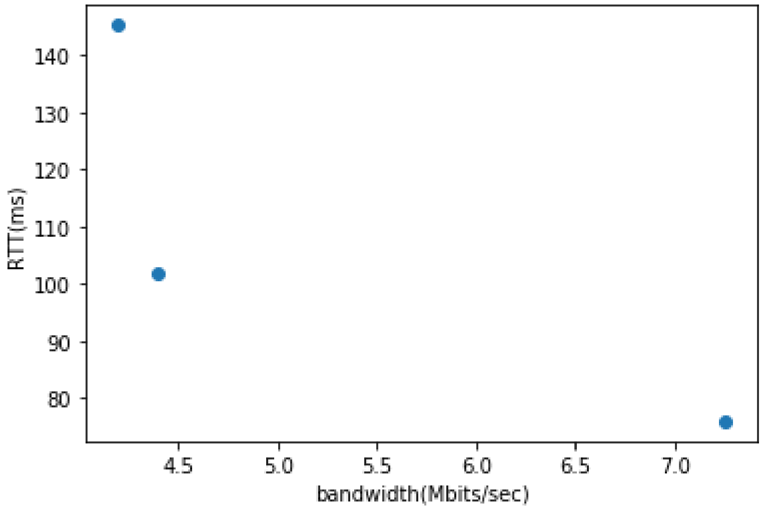
```sh
#!/bin/sh
TTL=255
TIME=10
COUNT=5
# France
echo "France"
ping bouygues.testdebit.info -t $TTL -c $COUNT > franceRTT.txt
iperf3 -c bouygues.testdebit.info -t $TIME -p 5204 > franceBand.txt

# Moscow
echo "Moscow"
ping speedtest.hostkey.ru -t $TTL -c $COUNT > MoscowRTT.txt
iperf3 -c speedtest.hostkey.ru -t $TIME -p 5201 > MoscowBand.txt

# US California
echo "california"
ping iperf.he.net -t $TTL -c $COUNT > us-caRTT.txt
iperf3 -c iperf.he.net -t $TIME -p 5201 > us-caBand.txt
```

Plot

```
RTT = [101.698, 145.248, 75.884]
bandwidth = [4.40, 4.20, 7.25]
# plt.plot(bandwidth, RTT, 'o', ls='-')
plt.scatter(bandwidth, RTT)
plt.xlabel('bandwidth(Mbits/sec)')
plt.ylabel('RTT(ms)')
plt.show()
```



Questions

1

> Describe the parameters you use, e.g., length of time to transmit (with -t), which servers you use (and where they are located), and other parameters if you specify any. Note, only the server names should vary, and all other parameters should be fixed.

For ping, I used `-t 255` and `-c 2`. For iperf3, I used `-t 10`.

2

> What mathematical relationship should you expect to see between throughput and RTT? Why?

They have a negative correlation, which makes sense, because a higher bandwith means means we can transfer data faster, so we should expect a lower RTT from that.

3

> If what you observe from your plot is not strictly the same as what's shown in the math equation (which is usually the case), then briefly explain what are other possible factors which may affect throughput.

Even though we expect a negative correlation between bandwidth and RTT, in many cases there will not be a perfect correlation. For example, a server far away with a high bandwidth may still result in a long RTT.

Wireshark

1. Curl/wget to grab a webpage:

Questions

> What's the command that you run?

`wget -O /dev/null https://jamesku.xyz`

> Any DNS packets?

Yes.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.0.95	75.75.75.75	DNS	71	Standard query 0x6f10 A jamesku.xyz
2	0.000017600	10.0.0.95	75.75.75.75	DNS	71	Standard query 0x1f1c AAAA jamesku.xyz
3	0.024033242	75.75.75.75	10.0.0.95	DNS	144	Standard query response 0x1f1c AAAA jamesku.xyz SOA dns1.p02.nsone.net
4	0.029145166	75.75.75.75	10.0.0.95	DNS	87	Standard query response 0x6f10 A jamesku.xyz A 104.248.63.231

> Describe the TCP packets that you see, i.e., how each packet corresponds to TCP handshake, data transfer and closing connection steps.

TCP handshake:

5	0.029418260	10.0.0.95	104.248.63.231	TCP	74	45712 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=489022082 TSecr=0 WS=128
6	0.053168993	104.248.63.231	10.0.0.95	TCP	70	443 → 45712 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=2447193058 TSecr=4890…
7	0.053213198	10.0.0.95	104.248.63.231	TCP	66	45712 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0 TSval=489022106 TSecr=2447193058

Application data and acks:

20	0.236345857	10.0.0.95	104.248.63.231	TLSv1.3	226	Application Data
21	0.301661964	104.248.63.231	10.0.0.95	TCP	66	443 → 45712 [ACK] Seq=3710 Ack=758 Win=31088 Len=0 TSval=2447193307 TSecr=489022289
22	0.332156711	104.248.63.231	10.0.0.95	TLSv1.3	466	Application Data
23	0.332174755	10.0.0.95	104.248.63.231	TCP	66	45712 → 443 [ACK] Seq=758 Ack=4110 Win=62738 Len=0 TSval=489022385 TSecr=2447193337

Closing connection:

70	0.431136090	10.0.0.95	104.248.63.231	TCP	66	45712 → 443 [FIN, ACK] Seq=758 Ack=165568 Win=65535 Len=0 TSval=489022484 TSecr=2447193434
71	0.456372183	104.248.63.231	10.0.0.95	TCP	66	443 → 45712 [FIN, ACK] Seq=165568 Ack=759 Win=31088 Len=0 TSval=2447193461 TSecr=489022484
72	0.456395787	10.0.0.95	104.248.63.231	TCP	66	45712 → 443 [ACK] Seq=759 Ack=165569 Win=65535 Len=0 TSval=489022509 TSecr=2447193461

2. Open a webpage in a browser

Questions

> Which website did you go to?

I visited my website: jamesku.xyz

> Do you see any parallel connections your browser makes? If so, how many can you see in your screenshot?

Yes. There are two connections. As we can see in the picture, there are two SYN packets.

34	1.144390748	10.0.0.95	151.139.128.10	TCP	74	35500 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1796987170 TSecr=0 WS=128
35	1.144580465	75.75.75.75	10.0.0.95	DNS	160	Standard query response 0x8695 AAAA kit.fontawesome.com SOA ns-832.awsdns-40.net
36	1.144895046	10.0.0.95	151.139.128.10	TCP	74	35502 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1796987171 TSecr=0 WS=128
37	1.160173578	151.139.128.10	10.0.0.95	TCP	74	443 → 35500 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=1218826793 TSecr=1796987170 WS=128
38	1.160212303	10.0.0.95	151.139.128.10	TCP	66	35500 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1796987186 TSecr=1218826793