



Hands-on: Traffic Control Challenge

Internet Engineering July 3rd 2019
By Teaching Assistant



Agenda

1. Setup Environment
2. Connect to Server
3. Customize Parameter



Download Makefile

1. Download makefile.

```
$ mkdir ~/Desktop/ieng/env && cd $_
```

```
$ curl -O
```

```
https://iplab.naist.jp/class/2019/materials/hands-on/04/makefile
```

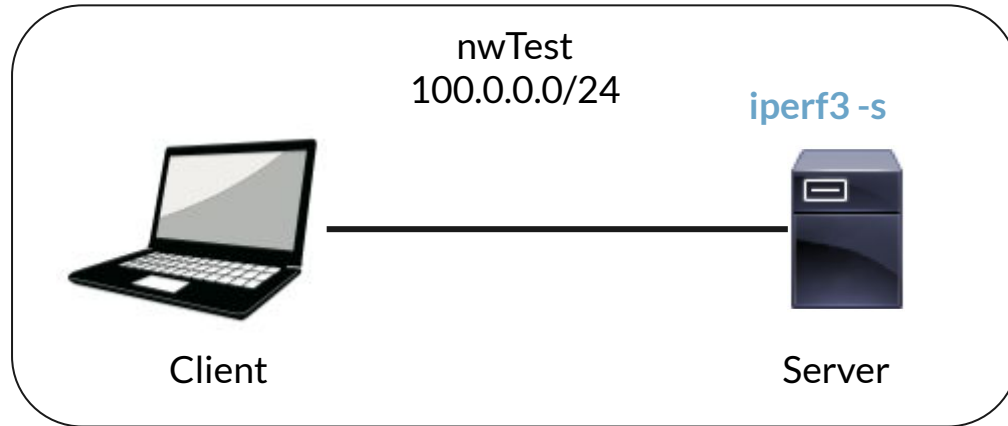
2. Remove previous containers for avoiding conflict.

```
$ make clean
```

3. Execute make command **on the same directory** as the downloaded makefile.

```
$ make
```

Network topology



iperf3 is running on the each servers.

'iperf3' is tool for active measurements of the maximum achievable bandwidth on IP networks.



Connect to Server from Client

(1) Attach client and server

```
$ docker attach client
```

```
$ docker attach server
```

(2) Confirm ip address of server

```
$ ip a | grep inet
```

(3) Launch wireshark on client

```
$ wireshark
```

(4) Connect to the server

```
$ iperf3 -c <Hostname or IP address>
```

Demo

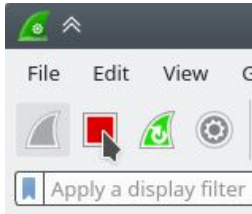


TCP Streams Graph

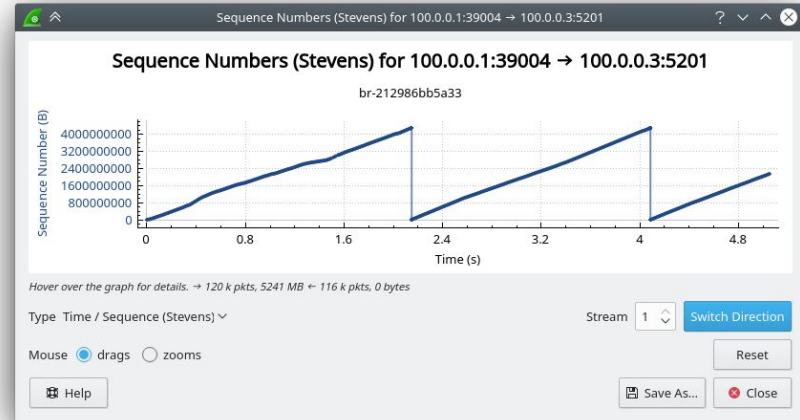
Anyway, let's capture the packet.

Wireshark has a useful tool called *TCP Streams Graph*.

1. Capture the packet
2. Stop the capture



3. [Statistics] -> [TCP Stream Graphs] -> [Time Sequences]



Manipulate Traffic Control Settings



The way to manipulate parameter

When you try to manipulate some parameters, use `tc` command on the server.

To see the current configuration:

```
$ tc qdisc show dev eth0
```

To change the configuration:

```
$ sudo tc qdisc change dev eth0 root netem [delay <N>ms | loss <N>%]
```

* There are another parameters, but we use `delay` and `loss` in this hands-on session.



Configuration: Delay

(1) Attach server.

```
$ docker attach server
```

(2) Add delay configuration with 3000ms

```
$ sudo tc qdisc add dev eth0 root netem delay 3000ms
```

(3) Confirm the current configuration:

```
$ tc qdisc show dev eth0
```

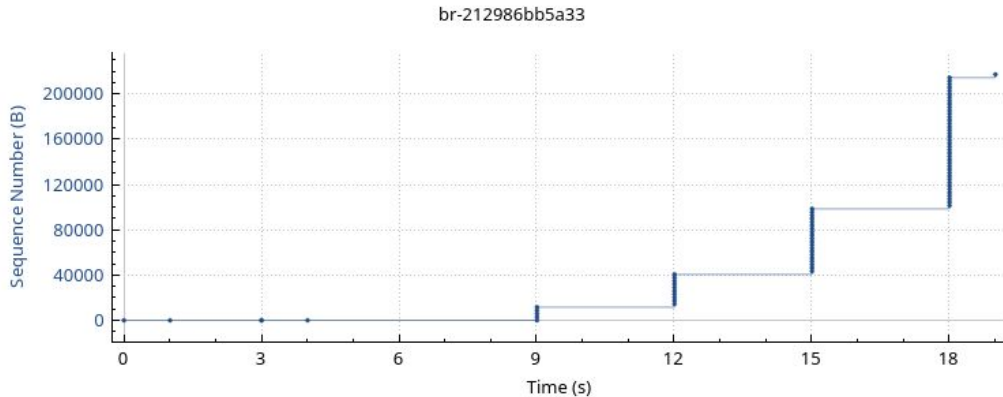
The result

As you see, we transfer the data only 294KBytes for 10 seconds.

Also, if you see TCP stream graph, sequence number increase every 3 seconds.

```
> iperf3 -c 100.0.0.3
Connecting to host 100.0.0.3, port 5201
[ 5] local 100.0.0.1 port 39132 connected to 100.0.0.3 port 5201
[ ID] Interval          Transfer          Bitrate          Retr  Cwnd
[ 5]  0.00-3.00    sec    107 KBytes      293 Kbits/sec     1   14.1 KBytes
[ 5]  3.00-4.00    sec     0.00 Bytes      0.00 bits/sec     0   14.1 KBytes
[ 5]  4.00-5.00    sec     0.00 Bytes      0.00 bits/sec     0   14.1 KBytes
[ 5]  5.00-6.00    sec    62.2 KBytes     510 Kbits/sec     0   28.3 KBytes
[ 5]  6.00-7.00    sec     0.00 Bytes      0.00 bits/sec     0   28.3 KBytes
[ 5]  7.00-8.00    sec     0.00 Bytes      0.00 bits/sec     0   28.3 KBytes
[ 5]  8.00-9.00    sec    124 KBytes     1.02 Mbits/sec     0   56.6 KBytes
[ 5]  9.00-10.00   sec     0.00 Bytes      0.00 bits/sec     0   56.6 KBytes
-----
[ ID] Interval          Transfer          Bitrate          Retr
[ 5]  0.00-10.00   sec    294 KBytes      241 Kbits/sec     1
[ 5]  0.00-13.00   sec    294 KBytes      62.4 Kbits/sec
iperf Done.
```

Sequence Numbers (Stevens) for 100.0.0.1:39132 → 100.0.0.3:5201





Configuration: Loss

(1) Attach server

```
$ docker attach server
```

(2) Add loss configuration with 50%

```
$ sudo tc qdisc add dev eth0 root netem loss 50%
```

(3) Confirm the current configuration

```
$ tc qdisc show dev eth0
```

The result

As you can see, it's happening TCP retransmissions(Reetr). Also, you can find the decline of data and bitrate.

```
ieng@ieng:~$ iperf3 -c 100.0.0.3
Connecting to host 100.0.0.3, port 5201
[ 5] local 100.0.0.2 port 42898 connected to 100.0.0.3 port 5201
[ ID] Interval           Transfer    Bitrate      Retr   Cwnd
[ 5]  0.00-1.00   sec    197 KBytes  1.61 Mbits/sec    1   36.8 KBytes
[ 5]  1.00-2.00   sec     0.00 Bytes  0.00 bits/sec     0   35.8 KBytes
[ 5]  2.00-3.00   sec    571 KBytes  4.67 Mbits/sec    4    1.41 KBytes
[ 5]  3.00-4.00   sec   1.10 MBytes  9.23 Mbits/sec    2   89.1 KBytes
[ 5]  4.00-5.00   sec    191 KBytes  1.56 Mbits/sec    2   102 KBytes
[ 5]  5.00-6.00   sec     0.00 Bytes  0.00 bits/sec    1    1.41 KBytes
[ 5]  6.00-7.00   sec     0.00 Bytes  0.00 bits/sec    1    1.41 KBytes
[ 5]  7.00-8.00   sec     0.00 Bytes  0.00 bits/sec    0    1.41 KBytes
[ 5]  8.00-9.01   sec     0.00 Bytes  0.00 bits/sec    0    1.41 KBytes
[ 5]  9.01-10.01  sec    191 KBytes  1.56 Mbits/sec    1   115 KBytes
-----
[ ID] Interval           Transfer    Bitrate      Retr
[ 5]  0.00-10.01  sec   2.22 MBytes  1.86 Mbits/sec   12
[ 5]  0.00-10.01  sec   2.09 MBytes  1.75 Mbits/sec
iperf Done.
```

Fin



Appendix



The way to manipulate parameter

To manipulate some parameters, use **iperf3** 's options on client side.

- Window size

```
$ iperf3 -c <hostname | ip address> -w <foo> [km]
```

- MSS(Maximum Segment Size)

```
$ iperf3 -c <hostname | ip address> -M <foo> [* default 1460ms]
```




Configure: tc

- Show the configuration

```
$ tc qdisc show dev <NIC>
```

- Delete the netem configuration

```
$ sudo tc qdisc del dev <NIC> root
```