Network Measurements

Manolis Surligas

surligas@csd.uoc.gr

Computer Science Department, University of Crete

Network Measurements

- Evaluate the performance of a network
- Spot hardware issues and bottlenecks
- Analyze topology
- Identify implementation bugs
- Stress test
- Cyberattack detection

Network Measurements

Metrics?

- Throughput
- Packet loss
- Latency
- Jitter
- Hop count

Network Measurements

More sophisticated metrics that maybe specific to the network technology or protocol

- Congestion window of TCP
- Re-transmissions
- CRC errors

Network Measurements

- The measurement metric used highly depends on the application
- Eg:
  - VoIP network measurements primary focus is on jitter and secondary on latency
  - Gaming network measurements may pay attention to latency
  - IPTV on throughput and packet loss

iperf3

The iperf3 tool

- Open source and cross platform
- Supports TCP, UDP and SCTP
- Client/Server operation
- Multiple connections support
- Generic throughput measurements

iperf3

Transport layer specific metrics and features

- TCP
  - Report MSS/MTU size and observed read sizes
  - TCP window size
- UDP
  - Measure packet loss
  - Measure jitter
  - UDP streams of specified bandwidth

iperf3 example

- Running the iperf server is as simple as:

```
comm8@comm8:~> iperf3 -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

- No need to specify the transport protocol or any other info as they are negotiated with the client through the control TCP secondary stream

iperf3 example

- On the client side, use the *-c* option to specify the server address
- For TCP:

```
surligas@localhost:> iperf3 -c 192.168.4.58
Connecting to host 192.168.4.58, port 5201
[  5] local 192.168.4.10 port 54408 connected to 192.168.4.58 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec   756 KBytes  6.19 Mbits/sec    0    150 KBytes
[  5]   1.00-2.00   sec  1.48 MBytes  12.4 Mbits/sec    0    262 KBytes
[  5]   2.00-3.00   sec  1.30 MBytes  10.9 Mbits/sec    0    311 KBytes
[  5]   3.00-4.00   sec   758 KBytes  6.21 Mbits/sec    8    301 KBytes
[  5]   4.00-5.00   sec   758 KBytes  6.21 Mbits/sec   21    255 KBytes
[  5]   5.00-6.00   sec   758 KBytes  6.21 Mbits/sec    0    298 KBytes
[  5]   6.00-7.00   sec   758 KBytes  6.21 Mbits/sec    0    319 KBytes
[  5]   7.00-8.00   sec  1.05 MBytes  8.80 Mbits/sec   21    233 KBytes
[  5]   8.00-9.00   sec   821 KBytes  6.73 Mbits/sec    0    258 KBytes
[  5]   9.00-10.00  sec   758 KBytes  6.21 Mbits/sec    0    269 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  9.07 MBytes  7.61 Mbits/sec   50             sender
[  5]   0.00-10.00  sec  7.89 MBytes  6.62 Mbits/sec                  receiver
```

iperf3 example

- For UDP, use the *-u* option
- **Do not forget to provide a desired bandwidth**

```
surligas@localhost:> iperf3 -c 192.168.4.58 -u -b 2M
Connecting to host 192.168.4.58, port 5201
[  5] local 192.168.4.10 port 51789 connected to 192.168.4.58 port 5201
[ ID] Interval           Transfer     Bitrate         Total Datagrams
[  5]   0.00-1.00   sec   245 KBytes  2.01 Mbits/sec  186
[  5]   1.00-2.00   sec   244 KBytes  2.00 Mbits/sec  185
[  5]   2.00-3.00   sec   245 KBytes  2.01 Mbits/sec  186
[  5]   3.00-4.00   sec   244 KBytes  2.00 Mbits/sec  185
[  5]   4.00-5.00   sec   245 KBytes  2.01 Mbits/sec  186
[  5]   5.00-6.00   sec   244 KBytes  2.00 Mbits/sec  185
[  5]   6.00-7.00   sec   245 KBytes  2.01 Mbits/sec  186
[  5]   7.00-8.00   sec   244 KBytes  2.00 Mbits/sec  185
[  5]   8.00-9.00   sec   244 KBytes  2.00 Mbits/sec  185
[  5]   9.00-10.00  sec   245 KBytes  2.01 Mbits/sec  186
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  5]   0.00-10.00  sec  2.38 MBytes  2.00 Mbits/sec  0.000 ms  0/1855 (0%)  sender
[  5]   0.00-10.00  sec  2.38 MBytes  2.00 Mbits/sec  3.874 ms  0/1854 (0%)  receiver
```

iperf3 and UDP

- Pay extreme attention on the throughput measurements during UDP experiments
- Trust only the receiver side
- Based on the parameters, the receiver maybe either the server or the client

iperf3 and UDP

- Let's try to conduct an experiment with 20 Mbps thoughput in a 10 Mbps capable connection
- The sender report may look like

```
surligas@localhost:> iperf3 -c 192.168.4.58 -u -b 20M
Connecting to host 192.168.4.58, port 5201
[  5] local 192.168.4.10 port 53331 connected to 192.168.4.58 port 5201
[ ID] Interval           Transfer     Bitrate         Total Datagrams
[  5]   0.00-1.00   sec  2.38 MBytes  20.0 Mbits/sec  1853
[  5]   1.00-2.00   sec  2.38 MBytes  20.0 Mbits/sec  1855
[  5]   2.00-3.00   sec  2.38 MBytes  20.0 Mbits/sec  1855
```

- Receiver side, tells a completely different story!

```
Accepted connection from 192.168.4.10, port 54496
[  5] local 192.168.4.58 port 5201 connected to 192.168.4.10 port 53331
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datagrams
[  5]   0.00-1.00   sec   925 KBytes  7.58 Mbits/sec  1.223 ms  648/1351 (48%)
[  5]   1.00-2.00   sec  1.13 MBytes  9.47 Mbits/sec  1.440 ms  1008/1886 (53%)
[  5]   2.00-3.00   sec  1.13 MBytes  9.49 Mbits/sec  0.647 ms  922/1802 (51%)
[  5]   3.00-4.00   sec  1.13 MBytes  9.46 Mbits/sec  1.051 ms  1048/1925 (54%)
```

iperf3 and UDP

- Why is this happening?
- Why this is not an issue for TCP?