Hands-on Session: iPerf3, NETEM, Bandwidth-Delay Product (BDP), TCP buffer size

Elie Kfoury, Jorge Crichigno
University of South Carolina
http://ce.sc.edu/cyberinfra

Hands-on Workshop on Networking Topics

April 5th, 12th, 2022

Content

- Introduction to Mininet
- Introduction to iPerf3
- Introduction to TCP buffers, BDP, and TCP window
- BDP and buffer size experiments
- Modifying buffer size and throughput test

NTP Lab Series

- Lab experiments

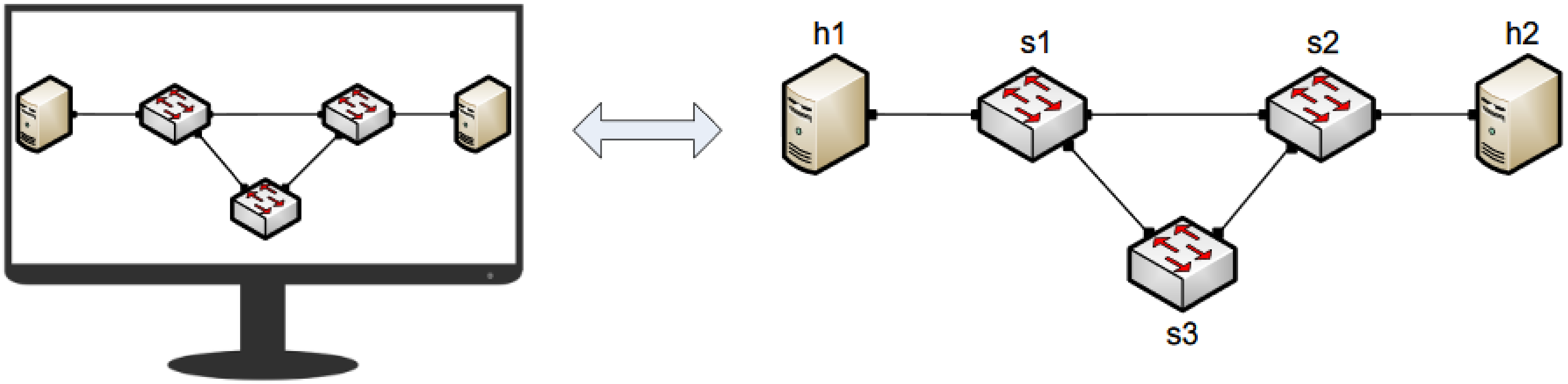
Lab 1: Introduction to Mininet
Lab 2: Introduction to iPerf
Lab 3: WANs with latency, Jitter
Lab 4: WANs with Packet Loss, Duplication, Corruption
Lab 5: Setting WAN Bandwidth with Token Bucket Filter (TBF)
Lab 6: Traditional TCP Congestion Control (HTCP, Cubic, Reno)
Lab 7: Rate-based TCP Congestion Control (BBR)
Lab 8: Bandwidth-delay Product and TCP Buffer Size
Lab 9: Enhancing TCP Throughput with Parallel Streams
Lab 10: Measuring TCP Fairness

Lab 11: Router's Buffer Size
Lab 12: TCP Rate Control with Pacing
Lab 13: Impact of Maximum Segment Size on Throughput
Lab 14: Router's Bufferbloat
Lab 15: Hardware Offloading on TCP Performance
Lab 16: Random Early Detection
Lab 17: Stochastic Fair Queueing
Lab 18: Controlled Delay (CoDel) Active Queue Management
Lab 19: Proportional Integral Controller-Enhanced (PIE)
Lab 20: Classifying TCP traffic using Hierarchical Token Bucket (HTB)

Introduction to Mininet

Mininet

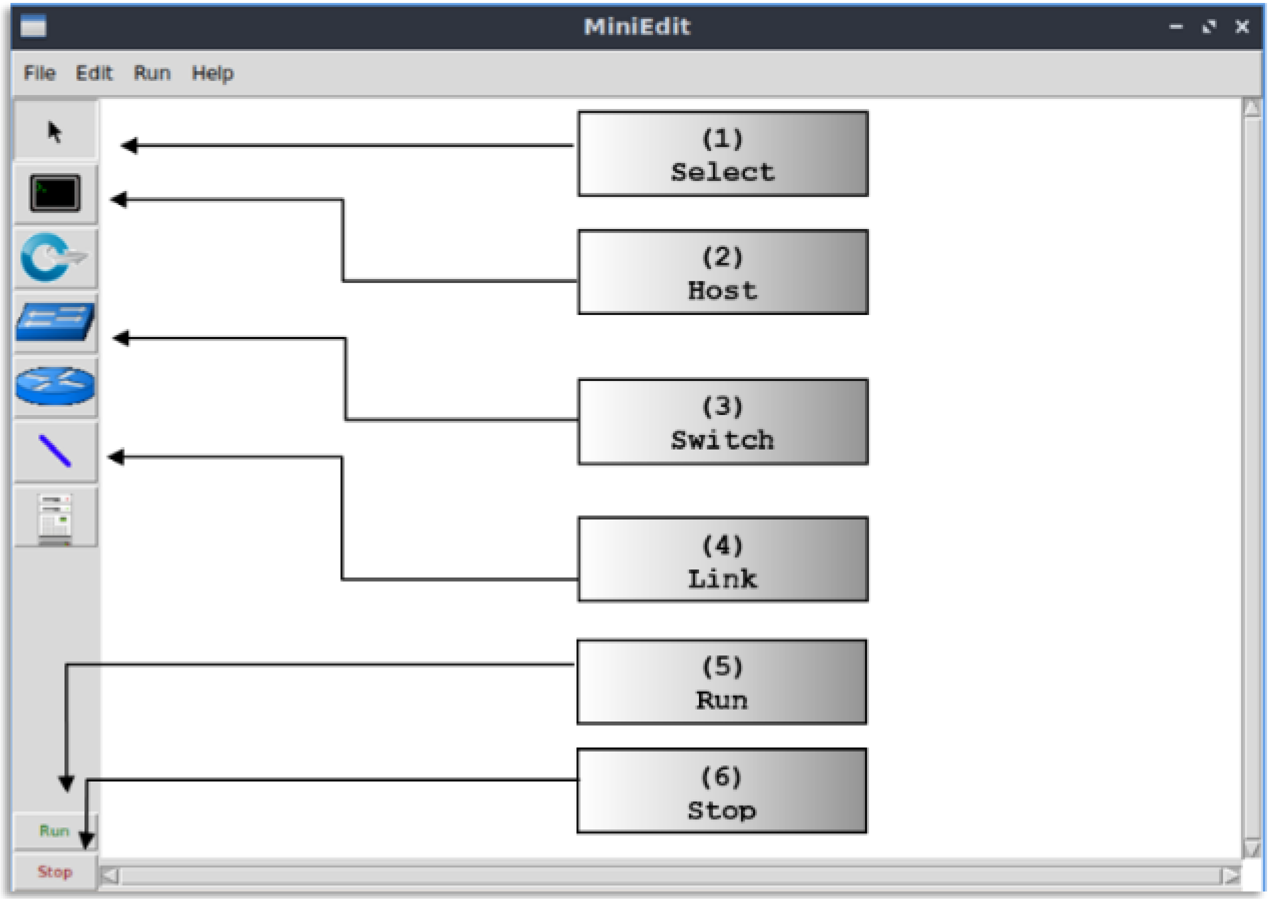
- Mininet provides network *emulation* opposed to simulation, allowing all network software at any layer to be simply run as is
- Mininet's logical nodes can be connected into networks
- Nodes are sometimes called containers, or more accurately, *network namespaces*
- Containers consume sufficiently few resources that networks of over a thousand nodes have been created, running on a single laptop



Mininet Emulated Network **Hardware Network**

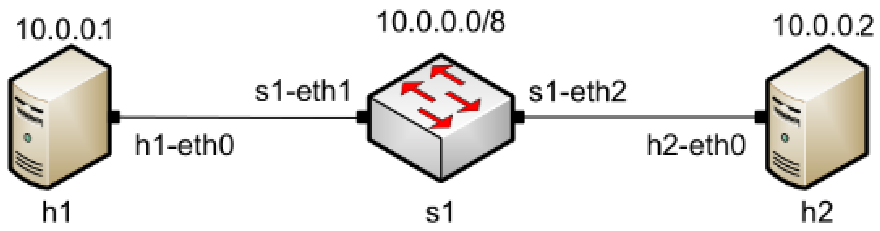
MiniEdit

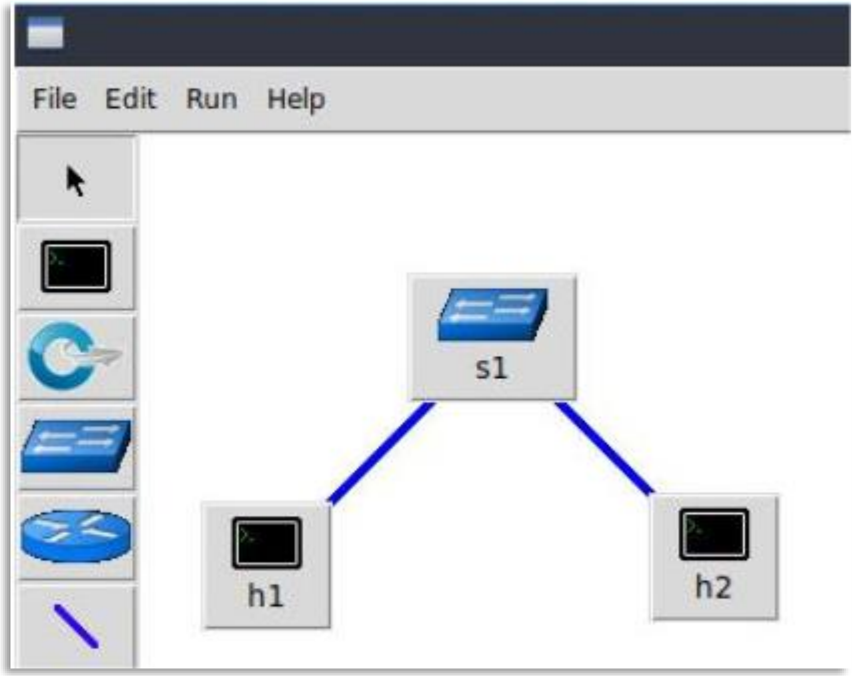
- MiniEdit is a simple GUI network editor for Mininet



MiniEdit

- To build Mininet’s minimal topology, two hosts and one switch must be deployed

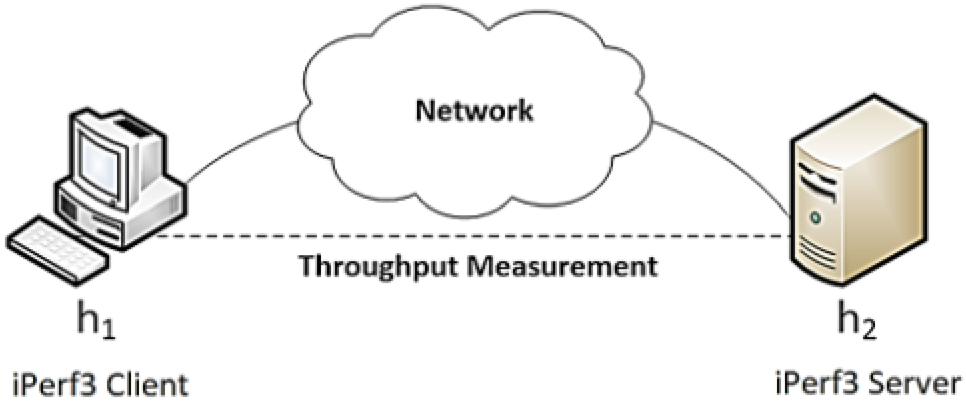




Introduction to iPerf3

iPerf3

- iPerf3 is a real-time network throughput measurement tool
- It is an open source, cross-platform client-server application that can be used to measure the throughput between the two end devices
- Measuring throughput is particularly useful when experiencing network bandwidth issues such as delay, packet loss, etc.



iPerf3

- iPerf3 can operate on TCP, UDP, and SCTP, unidirectional or bidirectional way
- In iPerf3, the user can set *client* and *server* configurations via options and parameters
- iPerf3 outputs a timestamped report of the amount of data transferred and the throughput measured

```
Connecting to host 10.0.0.2, port 5201
[ 13] local 10.0.0.1 port 59414 connected to 10.0.0.2 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[ 13]   0.00-1.00   sec  5.18 GBytes  44.5 Gbits/sec    0    843 KBytes
[ 13]   1.00-2.00   sec  5.21 GBytes  44.7 Gbits/sec    0   1.11 MBytes
[ 13]   2.00-3.00   sec  5.20 GBytes  44.7 Gbits/sec    0   1.18 MBytes
[ 13]   3.00-4.00   sec  5.21 GBytes  44.7 Gbits/sec    0   1.24 MBytes
[ 13]   4.00-5.00   sec  5.19 GBytes  44.6 Gbits/sec    0   1.24 MBytes
[ 13]   5.00-6.00   sec  5.22 GBytes  44.8 Gbits/sec    0   1.30 MBytes
[ 13]   6.00-7.00   sec  5.24 GBytes  45.0 Gbits/sec    0   1.44 MBytes
[ 13]   7.00-8.00   sec  5.22 GBytes  44.9 Gbits/sec    0   1.44 MBytes
[ 13]   8.00-9.00   sec  5.21 GBytes  44.8 Gbits/sec    0   1.45 MBytes
[ 13]   9.00-10.00  sec  5.22 GBytes  44.8 Gbits/sec    0   1.52 MBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[ 13]   0.00-10.00  sec  52.1 GBytes  44.8 Gbits/sec    0             sender
[ 13]   0.00-10.04  sec  52.1 GBytes  44.6 Gbits/sec                  receiver

iperf Done.
root@admin-pc:~#
```

Lab 8: Bandwidth-delay Product and TCP Buffer Size

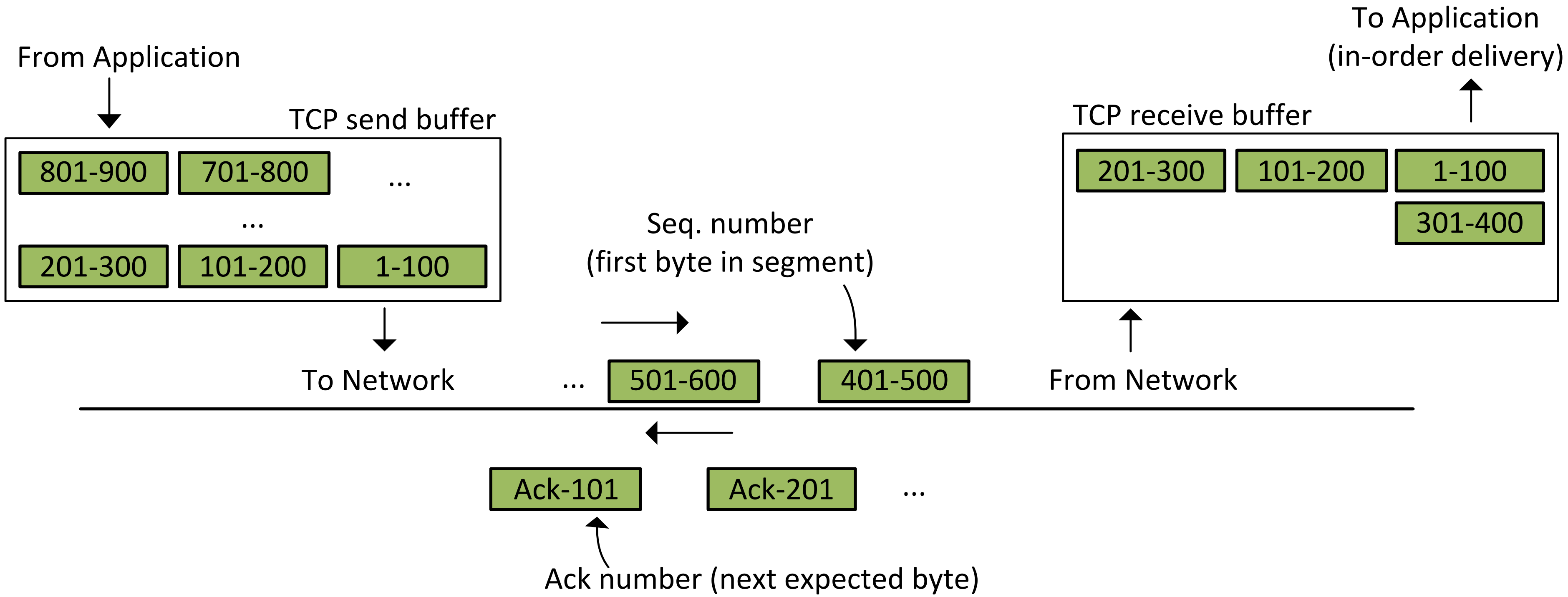
Section 1: Introduction to TCP buffers, BDP, and TCP window

TCP Buffers

- The TCP send and receive buffers may impact the performance of Wide Area Networks (WAN) data transfers
- At the sender side, TCP receives data from the application layer and places it in the TCP send buffer

TCP Buffers

- Typically, TCP fragments the data in the buffer into maximum segment size (MSS) units
- At any given time, the TCP receiver indicates the TCP sender how many bytes the latter can send, based on how much free buffer space is available at the receiver



TCP Buffers

- RTT and TCP buffer size have throughput implications
- For example, assume that the TCP buffer size is 1 Mbyte and RTT is 50ms
  - 1 Mbyte = 1,048,576 bytes = 1,048,576 · 8 bits = 8,388,608 bits
- With a bandwidth (Bw) of 10 Gbps, this number of bits is approximately transmitted in:

$$T_{tx} = \frac{\#\ \text{bits}}{\text{Bw}} = \frac{8{,}388{,}608}{10 \cdot 10^9} = 0.84\ \text{milliseconds.}$$

- After 0.84 milliseconds, the TCP send buffer will be empty
- TCP must wait for the corresponding acknowledgements (arriving at t = 50ms)
- This means that the sender only uses 0.84/50 or 1.68% of the available bandwidth

Bandwidth-delay product

- The solution lies in allowing the sender to continuously transmit segments until the corresponding acknowledgments arrive back
- The number of bits that can be transmitted in an RTT period is the bandwidth-delay product (BDP)
- For the previous example

$$\text{TCP buffer size} \geq \text{BDP} = \left(10 \cdot 10^{9}\right)\left(50 \cdot 10^{-3}\right) = 500{,}000{,}000 \text{ bits} = 62{,}500{,}000 \text{ bytes.}$$

- The first factor ($10 \cdot 10^{9}$) is the bandwidth; the second factor ($50 \cdot 10^{-3}$) is the RTT

$$\text{TCP buffer size} \geq 62{,}500{,}000 \text{ bytes} = 59.6 \text{ Mbytes} \approx 60 \text{ Mbytes.}$$

Practical Observations on Setting TCP Buffer Size

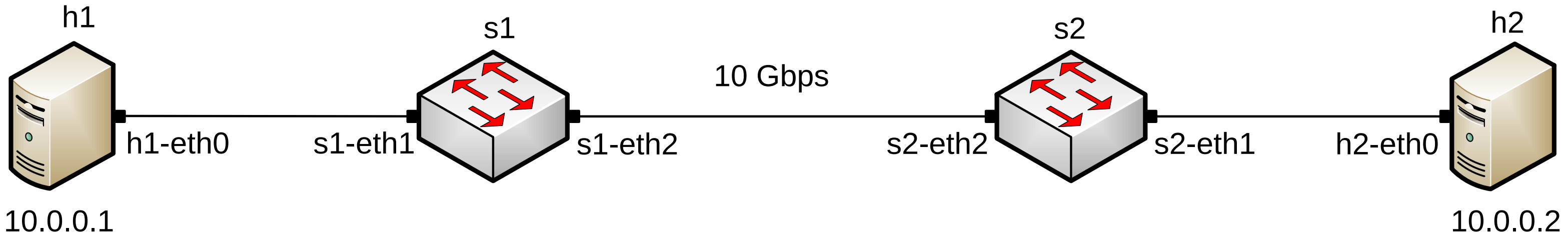
- Linux assumes that half of the send/receive TCP buffers are used for internal structures
- Thus, only half of the buffer size is used to store segments
- Considering the previous example, the TCP buffer size must be:

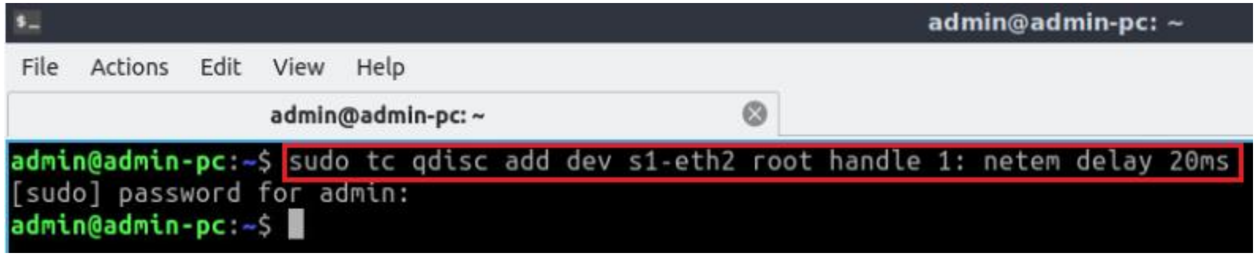
$$\text{TCP buffer size} \geq 2 \cdot 60 \text{ Mbytes} = 120 \text{ Mbytes}.$$

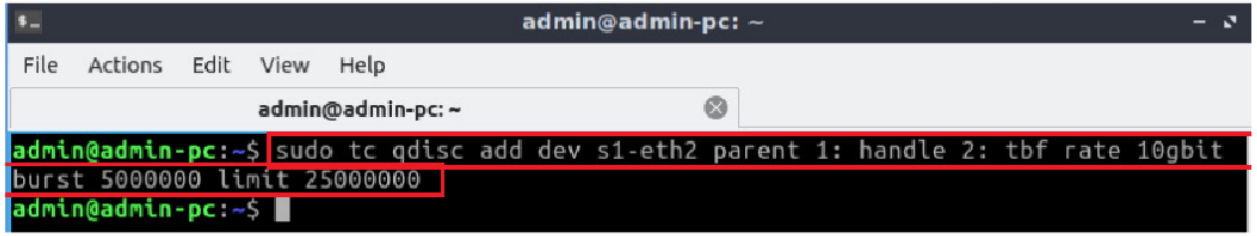
Section 2: BDP and buffer size experiments

Emulating a Wide Area Network

- The first figure shows the topology and the devices’ interfaces
- The second and third figures show the command that sets a latency of 20ms and bandwidth to 10 Gbps

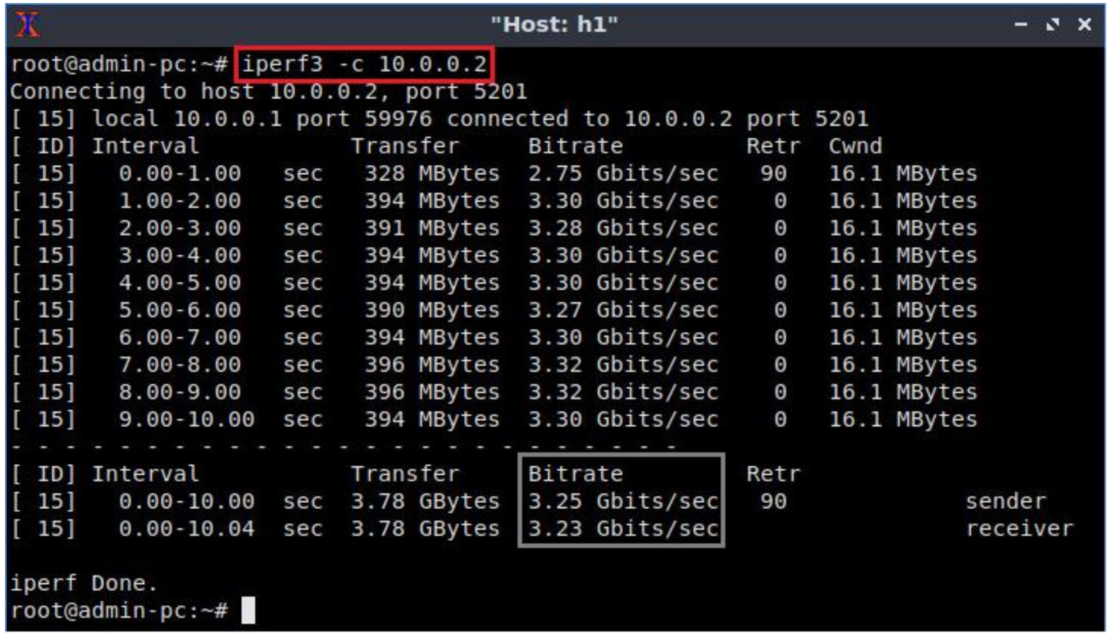




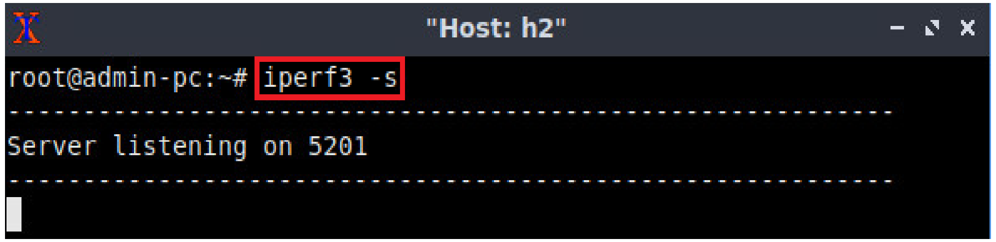


Verification

- The user can now verify the previous configuration by using the iperf3 tool to measure throughput



Client (h1)



Server (h2)

Section 3: Modifying buffer size and throughput test

BDP and buffer size

- To achieve the full throughput, the user has to modify the send and receive windows in host h1 and host h2

"Host: h1"

```
root@admin-pc:~# sysctl -w net.ipv4.tcp_rmem='10240 87380 52428800'
net.ipv4.tcp_rmem = 10240 87380 52428800
root@admin-pc:~#
```

"Host: h2"

```
root@admin-pc:~# sysctl -w net.ipv4.tcp_rmem='10240 87380 52428800'
net.ipv4.tcp_rmem = 10240 87380 52428800
root@admin-pc:~#
```

"Host: h1"

```
root@admin-pc:~# sysctl -w net.ipv4.tcp_wmem='10240 87380 52428800'
net.ipv4.tcp_wmem = 10240 87380 52428800
root@admin-pc:~#
```

"Host: h2"

```
root@admin-pc:~# sysctl -w net.ipv4.tcp_wmem='10240 87380 52428800'
net.ipv4.tcp_wmem = 10240 87380 52428800
root@admin-pc:~#
```

Verification

- The user can now verify the previous configuration by using the iperf3 tool to measure throughput

"Host: h1"

```
root@admin-pc:~# iperf3 -c 10.0.0.2
Connecting to host 10.0.0.2, port 5201
[ 15] local 10.0.0.1 port 47094 connected to 10.0.0.2 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[ 15]   0.00-1.00   sec   925 MBytes  7.76 Gbits/sec   45   39.8 MBytes
[ 15]   1.00-2.00   sec  1.11 GBytes  9.57 Gbits/sec    0   39.8 MBytes
[ 15]   2.00-3.00   sec  1.11 GBytes  9.56 Gbits/sec    0   39.8 MBytes
[ 15]   3.00-4.00   sec  1.11 GBytes  9.56 Gbits/sec    0   39.8 MBytes
[ 15]   4.00-5.00   sec  1.11 GBytes  9.56 Gbits/sec    0   39.8 MBytes
[ 15]   5.00-6.00   sec  1.11 GBytes  9.55 Gbits/sec    0   39.8 MBytes
[ 15]   6.00-7.00   sec  1.11 GBytes  9.56 Gbits/sec    0   39.8 MBytes
[ 15]   7.00-8.00   sec  1.11 GBytes  9.56 Gbits/sec    0   39.8 MBytes
[ 15]   8.00-9.00   sec  1.11 GBytes  9.56 Gbits/sec    0   39.8 MBytes
[ 15]   9.00-10.00  sec  1.11 GBytes  9.56 Gbits/sec    0   39.8 MBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[ 15]   0.00-10.00  sec  10.9 GBytes  9.38 Gbits/sec   45             sender
[ 15]   0.00-10.04  sec  10.9 GBytes  9.34 Gbits/sec                  receiver

iperf Done.
root@admin-pc:~#
```

Client (h1)

"Host: h2"

```
root@admin-pc:~# iperf3 -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

Server (h2)