Universidad de Puerto Rico
Recinto Universitario de Mayagüez
Engineering Faculty
Department of Computer Science

Project 4:

Francisco Diaz

802-15-2172

Prof. Kejie Lu

April 19 2020

Table of Content:

Overview

In this project we will measure bandwidth on IP networks. For that we will be using a performance analysis tool known as iperf and a graphical user interface for iperf known as jperf. The goal of this project is to become familiar with network performance analysis and interpret their results.

Link to Video

https://youtu.be/dPe3ltAfcX8

iPerf

iPerf is a compatible reimplementation of ttcp which also did network performance analysis.

iPerf 3 is the newest rewrite of iperf, the goal in doing so was to create a smaller code base and increase in functionality. Contrary to the previous version, iperf3 is single threaded. iPerf also has a graphical user interface known as jperf. Jperf is implemented in Java and extends certain functionality such as visual representations for bandwidth and command parameters.

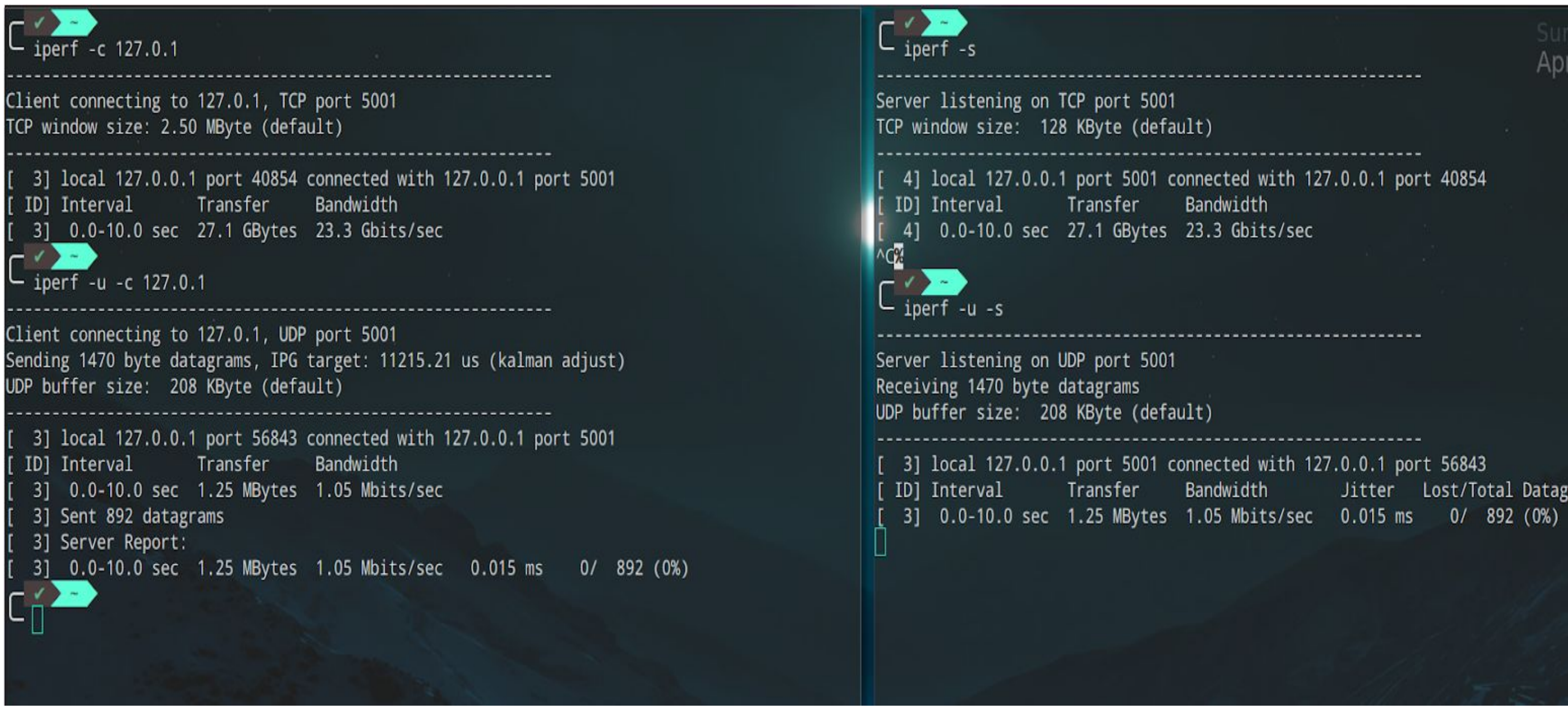
Where to Download and How to Execute

- I used a package manager to download iPerf
    - In Arch Linux it is: sudo pacman -S iperf
    - Binary is in /usr/bin/iperf path
    - Just run iperf

- For jperf:
    - https://sourceforge.net/projects/iperf/files/jperf/jperf%202.0.0/jperf-2.0.0.zip/download
    - Once downloaded unzip the compressed folder
    - Move to directory, add execute permission to the script using sudo chmod +x jperf.sh
    - Execute the script ./jperf.sh

Experiments in PC

- Commands
    - TCP
        - iperf -c 127.0.0.1
        - iperf -s
    - UDP
        - iperf -u -c 127.0.0.1
        - iperf -u -c 127.0.0.1
- Results

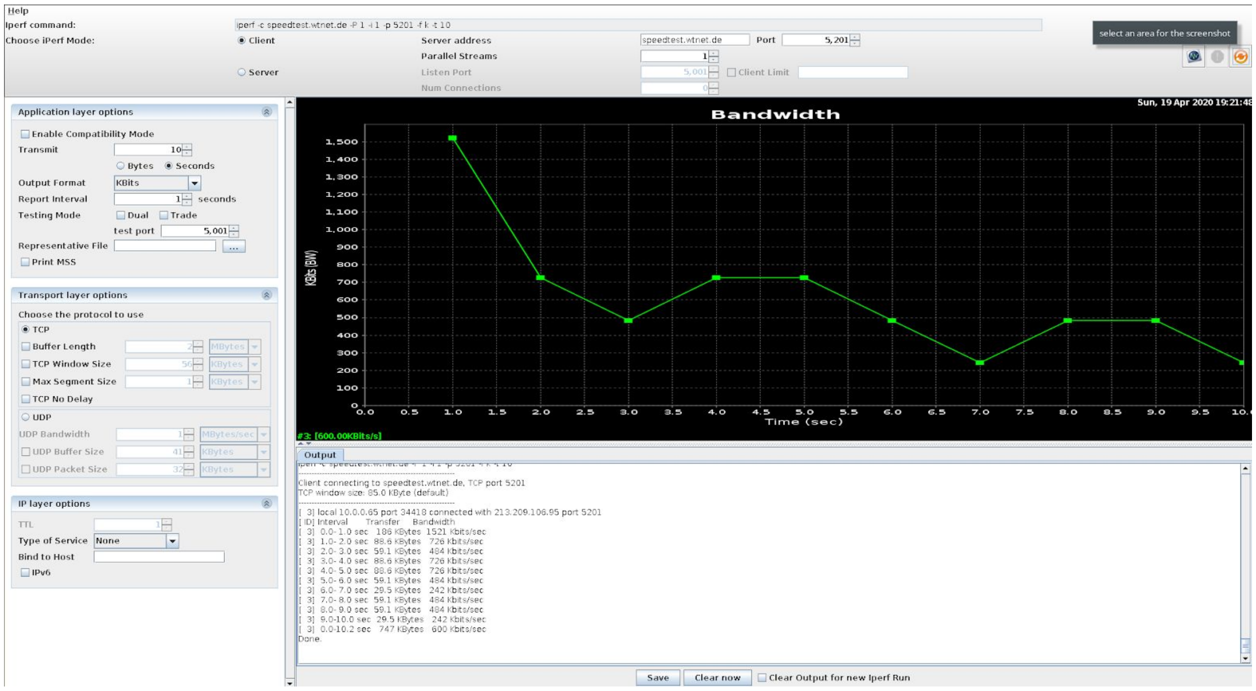


```
iperf -c 127.0.1
------------------------------------------------------------
Client connecting to 127.0.1, TCP port 5001
TCP window size: 2.50 MByte (default)
------------------------------------------------------------
[  3] local 127.0.0.1 port 40854 connected with 127.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  27.1 GBytes  23.3 Gbits/sec
iperf -u -c 127.0.1
------------------------------------------------------------
Client connecting to 127.0.1, UDP port 5001
Sending 1470 byte datagrams, IPG target: 11215.21 us (kalman adjust)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 127.0.0.1 port 56843 connected with 127.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec
[  3] Sent 892 datagrams
[  3] Server Report:
[  3]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec   0.015 ms    0/  892 (0%)
```

```
iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size:  128 KByte (default)
------------------------------------------------------------
[  4] local 127.0.0.1 port 5001 connected with 127.0.0.1 port 40854
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0-10.0 sec  27.1 GBytes  23.3 Gbits/sec
^C
iperf -u -s
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 127.0.0.1 port 5001 connected with 127.0.0.1 port 56843
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datag
[  3]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec  0.015 ms    0/  892 (0%)
```

Experiments with local server

- TCP


Help
Bandwidth
KBits (BW)
Time (sec)
Save
Clear now
Clear Output for new Iperf Run
select an area for the screenshot
Sun, 19 Apr 2020 19:21:48

Iperf command: iperf -c speedtest.wtnet.de -P 1 -i 1 -p 5201 -f k -t 10
Choose iPerf Mode: [✓ Client] Server
Server address: speedtest.wtnet.de   Port: 5,201
Parallel Streams: 1
Listen Port: 5,001   Client Limit
Num Connections: 0

Application layer options
Enable Compatibility Mode
Transmit: 10   Bytes [✓ Seconds]
Output Format: KBits
Report Interval: 1 seconds
Testing Mode: Dual Trade
test port: 5,001
Representative File
Print MSS

Transport layer options
Choose the protocol to use
[✓ TCP]
Buffer Length: 2 MBytes
TCP Window Size: 56 KBytes
Max Segment Size: 1 KBytes
TCP No Delay
UDP
UDP Bandwidth: 1 MBytes/sec
UDP Buffer Size: 41 KBytes
UDP Packet Size: 32 KBytes

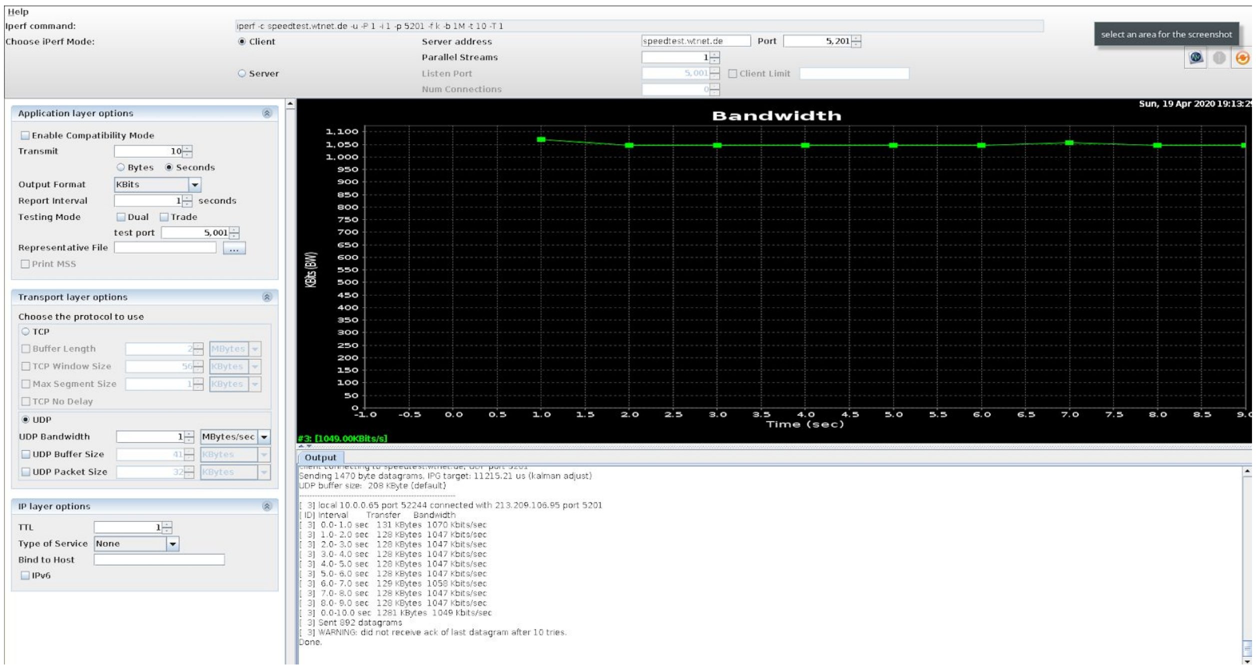
IP layer options
TTL: 1
Type of Service: None
Bind to Host
IPv6

#3: [600.00KBits/s]

Output

```
iperf -c speedtest.wtnet.de -P 1 -i 1 -p 5201 -f k -t 10
------------------------------------------------------------
Client connecting to speedtest.wtnet.de, TCP port 5201
TCP window size: 85.0 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.65 port 34418 connected with 213.209.106.95 port 5201
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec   186 KBytes  1521 Kbits/sec
[  3]  1.0- 2.0 sec  88.6 KBytes   726 Kbits/sec
[  3]  2.0- 3.0 sec  59.1 KBytes   484 Kbits/sec
[  3]  3.0- 4.0 sec  88.6 KBytes   726 Kbits/sec
[  3]  4.0- 5.0 sec  88.6 KBytes   726 Kbits/sec
[  3]  5.0- 6.0 sec  59.1 KBytes   484 Kbits/sec
[  3]  6.0- 7.0 sec  29.5 KBytes   242 Kbits/sec
[  3]  7.0- 8.0 sec  59.1 KBytes   484 Kbits/sec
[  3]  8.0- 9.0 sec  59.1 KBytes   484 Kbits/sec
[  3]  9.0-10.0 sec  29.5 KBytes   242 Kbits/sec
[  3]  0.0-10.2 sec   747 KBytes   600 Kbits/sec
Done.
```

- UDP



Iperf command: iperf -c speedtest.wtnet.de -u -P 1 -i 1 -p 5201 -f k -b 1M -t 10 -T 1

Choose iPerf Mode: [✓ Client] Server

Server address: speedtest.wtnet.de Port: 5,201

Parallel Streams: 1

Listen Port: 5,001 Client Limit

Num Connections: 0

Application layer options

- Enable Compatibility Mode
- Transmit: 10 Bytes [✓ Seconds]
- Output Format: KBits
- Report Interval: 1 seconds
- Testing Mode: Dual Trade
- test port: 5,001
- Representative File:
- Print MSS

Transport layer options

Choose the protocol to use: TCP [✓ UDP]

- Buffer Length: 2 MBytes
- TCP Window Size: 56 KBytes
- Max Segment Size: 1 KBytes
- TCP No Delay
- UDP Bandwidth: 1 MBytes/sec
- UDP Buffer Size: 41 KBytes
- UDP Packet Size: 32 KBytes

IP layer options

- TTL: 1
- Type of Service: None
- Bind to Host:
- IPv6

```
Sending 1470 byte datagrams, IPG target: 11215.21 us (kalman adjust)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.65 port 52244 connected with 213.209.106.95 port 5201
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec   131 KBytes  1070 Kbits/sec
[  3]  1.0- 2.0 sec   128 KBytes  1047 Kbits/sec
[  3]  2.0- 3.0 sec   128 KBytes  1047 Kbits/sec
[  3]  3.0- 4.0 sec   128 KBytes  1047 Kbits/sec
[  3]  4.0- 5.0 sec   128 KBytes  1047 Kbits/sec
[  3]  5.0- 6.0 sec   128 KBytes  1047 Kbits/sec
[  3]  6.0- 7.0 sec   129 KBytes  1058 Kbits/sec
[  3]  7.0- 8.0 sec   128 KBytes  1047 Kbits/sec
[  3]  8.0- 9.0 sec   128 KBytes  1047 Kbits/sec
[  3]  0.0-10.0 sec  1281 KBytes  1049 Kbits/sec
[  3] Sent 892 datagrams
[  3] WARNING: did not receive ack of last datagram after 10 tries.
Done.
```

Experiments with remote server (TCP)

Port Number	speedtest.wtnet.de	ping.online.net	speedtest.serverius.net
5200	651 Kbits/sec	587 Kbits/sec	Connection failed
5201	654 Kbits/sec	Broken pipe	63.2 Kbits/sec
5202	675 Kbits/sec	605 Kbits/sec	Broken Pipe

Experiments with remote server (UDP)

Port Number	speedtest.wtnet.de	ping.online.net	speedtest.serverius.net
5200	1049 Kbits/sec	1049 Kbits/sec	1049 Kbits/sec
5201	1049 Kbits/sec	1049 Kbits/sec	1049 Kbits/sec
5202	1049 Kbits/sec	1049 Kbits/sec	1049 Kbits/sec