

PLANIFICACIÓN E INTEGRACIÓN DE SISTEMAS Y SERVICIOS

Práctica 1

Pablo Alcázar Morales

Juan Manuel Palacios Navas

Apartado 1

1. Ancho de banda:

El ancho de banda indica el rango neto de bits en un sistema de comunicación digital

2. Latencia

La latencia es la suma de retardos temporales dentro de una red, producidos por la demora en la propagación y transmisión de paquete de la red.

3. RTT

Es el tiempo que tarda un paquete en ir y volver, desde que un emisor lo envía hasta que recibe una respuesta.

4. Jitter

Es la variación de tiempo en la llegada de los paquetes, causada por la congestión de la red, o por las diferentes rutas que siguen los paquetes para llegar al destino.

5. Throughput

Es la velocidad real de un canal de comunicación.

6. Perdida de datagramas.

Si usamos el protocolo UDP, nuestros paquetes serán llamados datagramas. Dadas las características asíncronas del protocolo UDP, es posible que se pierdan datagramas, y es una característica que también debería medirse para asegurar la calidad de una conexión.

Apartado 2

• Loopback

En TCP, sin cambiar el tamaño de la ventana, podemos observar como el Throughput se mantiene entorno a los 47 GB/s. Si cambiamos la ventana, se reduce proporcionalmente.

```
vagrant@ubuntu-bionic:~$ iperf -c 127.0.0.1
-----
Client connecting to 127.0.0.1, TCP port 5001
TCP window size: 2.50 MByte (default)
-----
[ 3] local 127.0.0.1 port 39146 connected with 127.0.0.1 port 5001
[ 4] local 127.0.0.1 port 5001 connected with 127.0.0.1 port 39146
[ ID] Interval      Transfer    Bandwidth
[ 3]  0.0-10.0 sec  48.0 GBytes 41.2 Gbits/sec
[ ID] Interval      Transfer    Bandwidth
[ 4]  0.0-10.0 sec  48.0 GBytes 41.2 Gbits/sec
```

En UDP, la tasa se reduce a 1.25 MB/s

```

iperf -c 127.0.0.1 -u
-----
Client connecting to 127.0.0.1, 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 58684 connected with 127.0.0.1 port 5001
[ 3] local 127.0.0.1 port 5001 connected with 127.0.0.1 port 58684
[ ID] Interval      Transfer    Bandwidth      Jitter    Lost/Total Datagrams
[ 3] 0.0-10.0 sec  1.25 MBytes 1.05 Mbits/sec  0.008 ms   0/ 893 (0%)
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.0 sec  1.25 MBytes 1.05 Mbits/sec
[ 3] Sent 893 datagrams
[ 3] Server Report:
[ 3] 0.0-10.0 sec  1.25 MBytes 1.05 Mbits/sec  0.000 ms   0/ 893 (0%)

```

- **Dos interfaces distintas**

Optenemos aproximadamente los mismo resultado que en la interfaz loopback.

En TCP:

```

vagrant@ubuntu-bionic:~$ iperf -c 100.100.1.2
[ 4] local 100.100.1.2 port 5001 connected with 100.100.1.2 port 39256
-----
Client connecting to 100.100.1.2, TCP port 5001
TCP window size: 2.50 MByte (default)
-----
[ 3] local 100.100.1.2 port 39256 connected with 100.100.1.2 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.0 sec  46.9 GBytes 40.2 Gbits/sec
[ ID] Interval      Transfer    Bandwidth
[ 4] 0.0-10.0 sec  46.9 GBytes 40.2 Gbits/sec

```

Y en UDP:

```

vagrant@ubuntu-bionic:~$ iperf -c 100.100.1.2 -u
-----
Client connecting to 100.100.1.2, UDP port 5001
[ 3] local 100.100.1.2 port 5001 connected with 100.100.1.2 port 40045
Sending 1470 byte datagrams, IPG target: 11215.21 us (kalman adjust)
UDP buffer size: 208 KByte (default)
-----
[ 3] local 100.100.1.2 port 40045 connected with 100.100.1.2 port 5001
[ ID] Interval      Transfer    Bandwidth      Jitter    Lost/Total Datagrams
[ ID] Interval      Transfer    Bandwidth      Jitter    Lost/Total Datagrams
[ 3] 0.0-10.0 sec  1.25 MBytes 1.05 Mbits/sec  0.002 ms   0/ 893 (0%)
[ 3] 0.0-10.0 sec  1.25 MBytes 1.05 Mbits/sec
[ 3] Sent 893 datagrams
[ 3] Server Report:
[ 3] 0.0-10.0 sec  1.25 MBytes 1.05 Mbits/sec  0.000 ms   0/ 893 (0%)

```

- **Dos maquinas distintas**

En TCP la calidad de la conexión se reduce notablemente, como podemos ver en la imagen

```

Client connecting to 10.0.3.21, TCP port 5001
TCP window size: 85.0 KByte (default)
-----
[ 3] local 10.0.3.20 port 46656 connected with 10.0.3.21 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.0 sec  2.45 GBytes 2.10 Gbits/sec

```

Al igual que en el resto de casos, si reduces el tamaño de la ventana, se reduce el throughput.

```
vagrant@ubuntu-bionic:~$ iperf -c 10.0.3.21 -w 1K
WARNING: TCP window size set to 1024 bytes. A small window size
will give poor performance. See the Iperf documentation.
-----
Client connecting to 10.0.3.21, TCP port 5001
TCP window size: 4.50 KByte (WARNING: requested 1.00 KByte)
-----
[ 3] local 10.0.3.20 port 52320 connected with 10.0.3.21 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0-10.0 sec   392 MBytes    329 Mbits/sec
```

Con conexiones paralelas, igualmente, se divide el throughput, pero la suma de ellos es igual a cuando se hace solo una conexión.

```
[ ID] Interval      Transfer      Bandwidth
[ 4] 0.0-10.0 sec   165 MBytes    138 Mbits/sec
[ 7] 0.0-10.0 sec   167 MBytes    140 Mbits/sec
[ 8] 0.0-10.0 sec   178 MBytes    149 Mbits/sec
[ 9] 0.0-10.0 sec   166 MBytes    139 Mbits/sec
[12] 0.0-10.0 sec   184 MBytes    154 Mbits/sec
[ 5] 0.0-10.0 sec   174 MBytes    146 Mbits/sec
[ 6] 0.0-10.0 sec   152 MBytes    127 Mbits/sec
[10] 0.0-10.0 sec   173 MBytes    145 Mbits/sec
[11] 0.0-10.0 sec   165 MBytes    138 Mbits/sec
[ 3] 0.0-10.0 sec   174 MBytes    146 Mbits/sec
[SUM] 0.0-10.0 sec 1.66 GBytes    1.42 Gbits/sec
```

- **Cuello de botella**

Asumiendo que lo que nos puede interesar es tener un throughput alto, la ecuación para calcularlo es la siguiente:

[TCP Window Size in bits] / [Latency in seconds] = [Throughput in bits per second]

Por tanto, dado que no podemos controlar la latencia, el cuello de botella vendrá determinado por el tamaño de la ventana

- **Conclusiones**

Con iperf podemos conocer los límites de nuestra red, y que repercusiones tendrían en ella la modificación de algunas características, como el tamaño de la ventana.

Aun así esta simulación es bastante hipotética, ya que depende de dos máquinas distintas, y del proceso que este actuando a modo de servidor, que en este caso se entiende que es bastante sencillo (consume pocos recursos).