

## Trabajo Práctico #4

# Calidad de Servicio QoS

Fecha de Entrega: 13-10-2016

Medina, Ma Victoria [117024]

medina.vicc@gmail.com

- 1) Se requiere configurar una política de control de tráfico para el uso de un enlace de red que limite la tasa de transferencia independientemente del tipo de tráfico cursado por el enlace. Para este trabajo es necesario configurar una PC como router donde se aplica la política y un cliente y servidor para la generación del tráfico (por ejemplo, puede utilizar la herramienta nc). Detalle la configuración realizada para permitir ruteo y la configuración de la política de control aplicada en cada una de las interfaces de red. Verifique la aplicación de la política utilizando las herramientas de medición de tráfico vistas en clase (Por ejemplo mediante iperf).

Configuración del router:

```
sysch net.ipv4.ip_forward=1 (para hacer ruteo)
```

Luego se borraron las interfaces que tenían ya establecidas:

```
ip addr del 10.4.10.32 dev eth0
```

Se establece como ruta por defecto la maquina .30 :

```
ip route default 10.4.10.30
```

Y se asignaron las direcciones para cada interfaz:

```
ip addr add 10.5.0.1 24 dev eth0
```

```
ip addr add 10.6.0.1 24 dev eth0
```

```
root@Redes:/home/alumno# ip route add default via 10.5.0.100 dev eth0  
#agregamos la ruta por defecto
```

Se configura la política:

```
root@Redes:/home/alumno# tc qdisc add dev eth0 root tbf rate 1mbit burst 100kb latency 50ms
```

Se inicia como servidor para limitar la velocidad en que me llegan los datos:

```
root@Redes:/home/alumno# iperf -s
```

```
-----  
Server listening on TCP port 5001  
TCP window size: 85.3 KByte (default)  
-----
```

```
[ 4] local 10.5.0.1 port 5001 connected with 10.6.0.1 port 55588  
[ ID] Interval          Transfer      Bandwidth  
[ 4] 0.0-12.0 sec    1.38 MBytes    963 Kbits/sec
```

## Trabajo Práctico #4

# Calidad de Servicio QoS

Fecha de Entrega: 13-10-2016

Medina, Ma Victoria [117024]

medina.vicc@gmail.com

Ahora al revés, yo como servidor:

```
root@Redes:/home/alumno# iperf -s
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[  4] local 10.5.0.1 port 5001 connected with 10.6.0.1 port 55588
[ ID] Interval          Transfer      Bandwidth
[  4]  0.0-12.0 sec    1.38 MBytes    963 Kbits/sec

root@Redes:/home/alumno# iperf -c 10.6.0.1
-----
Client connecting to 10.6.0.1, TCP port 5001
TCP window size: 43.8 KByte (default)
-----
[  3] local 10.5.0.1 port 32963 connected with 10.6.0.1 port 5001
[ ID] Interval          Transfer      Bandwidth
[  3]  0.0-10.0 sec    63.6 MBytes    53.3 Mbits/sec
```

- 2) Una organización requiere configurar una política de control de tráfico que priorice los servicios de correo electrónico, web y curse el resto del tráfico sin prioridad. El enlace existente en la organización es de 2 Mbps, y se requiere que el correo pueda llegar a operar hasta 1.5 Mbps y el tráfico web hasta 1 Mbps de tasa de transferencia. Proponga una solución utilizando algunas de las disciplinas de encolado vistas. Describa someramente su implementación y válidelas con las herramientas de monitoreo ya utilizadas, generando tráfico sintético para los puertos de los servicios involucrados. Documente las correspondientes salidas.

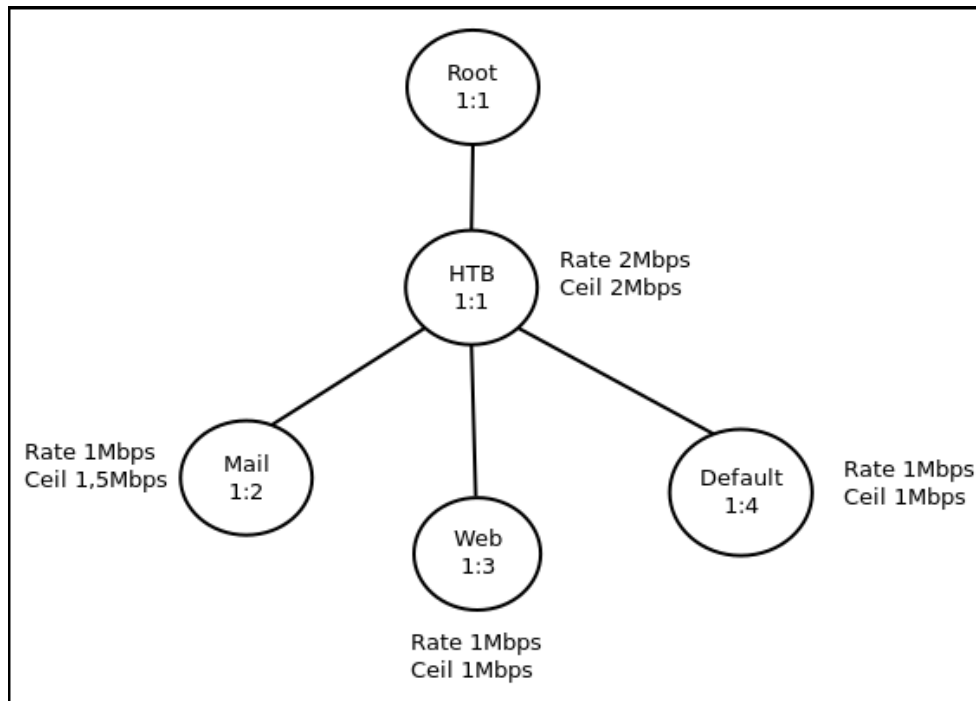
## Trabajo Práctico #4

# Calidad de Servicio QoS

Fecha de Entrega: 13-10-2016

Medina, Ma Victoria [117024]

medina.vicc@gmail.com



Borra todas las disciplinas: `tc qdisc del dev eth0 root`

`-tc qdisc add dev eth0 root handle 1: htb default 6`

A la interfaz le quita la disciplina y le pone una del tipo htb (hierarchical token bucket) para poder configurar. Esto se lo hace al root.

Definimos disciplina de encolado:

`-tc class add dev eth0 parent 1:1 classid 1:1 htb rate 2mbit ceil 2mbps`

Se define la clase 1:1 con 2 mb de base y techo. Y de esta van a salir las otras clases derivadas.

`-tc class add dev eth0 parent 1:1 classid 1:2 htb rate 1mbit ceil 1.5mbps`

Se crea la clase del Servidor web

`-tc class add dev eth0 parent 1:1 classid 1:6 htb rate 0.5mbit ceil 1mbit`

Se crea la clase 1:6

`-tc filter add dev eth0 protocol ip parent 1:0 prio 0 u32 match ip dport 80 0xffff flowid 1:3`

## Trabajo Práctico #4

# Calidad de Servicio QoS

Fecha de Entrega: 13-10-2016

Medina, Ma Victoria [117024]

medina.vicc@gmail.com

```
-tc filter add dev eth0 protocol ip parent 1:0 prio 0 u32 match ip  
src idp port 80 0xffff flowid 1:3
```

3) Realice una simulación de una red de área amplia (WAN) donde se evalúen las siguientes cuestiones:

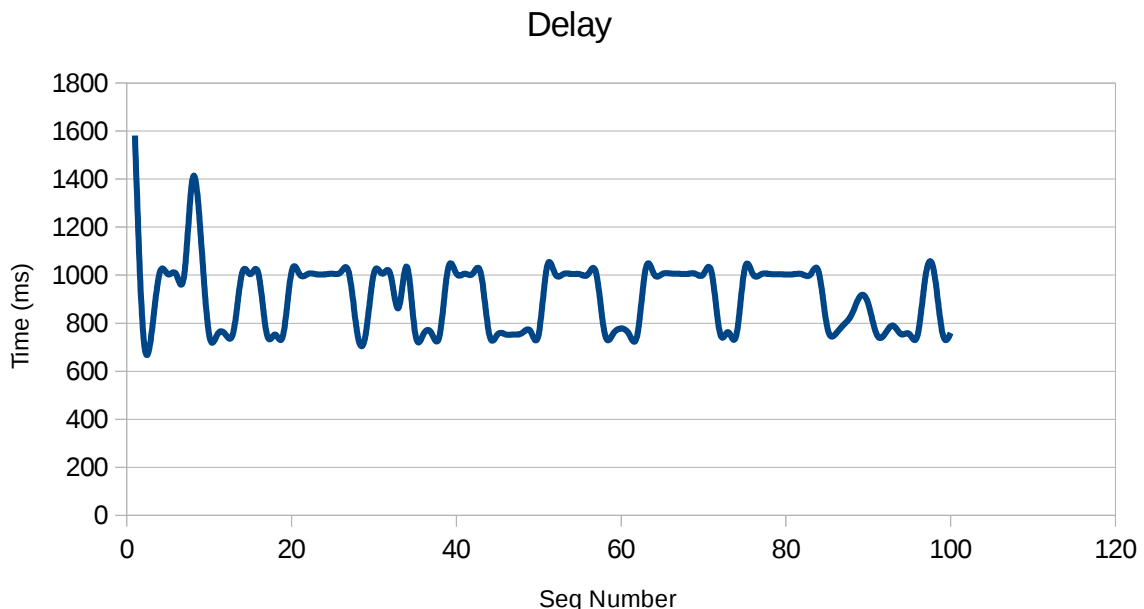
- Delay variable: Aplique una regla de delay, definiendo el valor que este adoptará, su variación y un valor de correlación. Detalle la configuración establecida. Evalúe empíricamente el delay existente y grafique el comportamiento de la regla establecida.

En el router se definió la siguiente política:

```
root@viccm-C500:/home/viccm# tc qdisc add dev wlpls0 root netem delay 750ms
```

En el cliente se procedió a realizar el ping con un máximo de 100 para obtener valores representativos de la situación:

```
PING 192.168.0.108 (192.168.0.108) 56(84) bytes of data.  
--- 192.168.0.108 ping statistics ---  
100 packets transmitted, 100 received, 0% packet loss, time 99239ms  
rtt min/avg/max/mdev = 752.919/912.114/1581.740/148.393 ms, pipe 2
```



En el gráfico se puede observar como la transmisión sufre variaciones en los tiempos en que se envía cada secuencia.

**Trabajo Práctico #4**  
**Calidad de Servicio QoS**  
**Fecha de Entrega: 13-10-2016**  
Medina, Ma Victoria [117024]  
medina.vicc@gmail.com

- b) Pérdida: Simule un escenario de pérdida de paquetes, luego utilice la herramienta ping para verificar que efectivamente las reglas fueron correctamente definidas en el host. Detalle la configuración establecida. Realice una captura para visualizar el comportamiento y documente lo hallado.

Se definió la siguiente política:

```
root@viccm-C500:/home/viccm# tc qdisc replace dev wlp1s0 root netem loss 25%
```

Time	Source	Destination	Protocol	Length	Info
49 4.844223...	192.168.0.108	192.168.0.106	ICMP	98	Echo (ping) request id=0x137f, seq=1/256, ttl=64 (no response found!)
59 5.852474...	192.168.0.108	192.168.0.106	ICMP	98	Echo (ping) request id=0x137f, seq=2/512, ttl=64 (no response found!)
71 6.860453...	192.168.0.108	192.168.0.106	ICMP	98	Echo (ping) request id=0x137f, seq=3/768, ttl=64 (no response found!)
82 7.868440...	192.168.0.108	192.168.0.106	ICMP	98	Echo (ping) request id=0x137f, seq=4/1024, ttl=64 (no response found!)
97 8.876496...	192.168.0.108	192.168.0.106	ICMP	98	Echo (ping) request id=0x137f, seq=5/1280, ttl=64 (no response found!)
110 9.884443...	192.168.0.108	192.168.0.106	ICMP	98	Echo (ping) request id=0x137f, seq=6/1536, ttl=64 (no response found!)
122 10.89243...	192.168.0.108	192.168.0.106	ICMP	98	Echo (ping) request id=0x137f, seq=7/1792, ttl=64 (no response found!)
134 11.90053...	192.168.0.108	192.168.0.106	ICMP	98	Echo (ping) request id=0x137f, seq=8/2048, ttl=64 (no response found!)
163 14.07240...	192.168.0.106	192.168.0.108	ICMP	98	Echo (ping) reply id=0x137f, seq=2/512, ttl=64 (request in 59)
164 14.09533...	192.168.0.106	192.168.0.108	ICMP	98	Echo (ping) reply id=0x137f, seq=3/768, ttl=64 (request in 71)
165 14.09536...	192.168.0.106	192.168.0.108	ICMP	98	Echo (ping) reply id=0x137f, seq=4/1024, ttl=64 (request in 82)
166 14.09537...	192.168.0.106	192.168.0.108	ICMP	98	Echo (ping) reply id=0x137f, seq=5/1280, ttl=64 (request in 97)
167 14.09537...	192.168.0.106	192.168.0.108	ICMP	98	Echo (ping) reply id=0x137f, seq=5/1280, ttl=64
169 14.09539...	192.168.0.108	192.168.0.106	ICMP	98	Echo (ping) request id=0x137f, seq=9/2304, ttl=64 (reply in 176)
170 14.09542...	192.168.0.108	192.168.0.106	ICMP	98	Echo (ping) request id=0x137f, seq=10/2560, ttl=64 (reply in 177)
171 14.09546...	192.168.0.106	192.168.0.108	ICMP	98	Echo (ping) reply id=0x137f, seq=6/1536, ttl=64 (request in 110)
173 14.09547...	192.168.0.106	192.168.0.108	ICMP	98	Echo (ping) reply id=0x137f, seq=7/1792, ttl=64 (request in 122)
175 14.09548...	192.168.0.106	192.168.0.108	ICMP	98	Echo (ping) reply id=0x137f, seq=8/2048, ttl=64 (request in 134)
176 14.10281...	192.168.0.106	192.168.0.108	ICMP	98	Echo (ping) reply id=0x137f, seq=9/2304, ttl=64 (request in 169)
177 14.10696...	192.168.0.106	192.168.0.108	ICMP	98	Echo (ping) reply id=0x137f, seq=10/2560, ttl=64 (request in 170)
187 14.91797...	192.168.0.108	192.168.0.106	ICMP	98	Echo (ping) request id=0x137f, seq=11/2816, ttl=64 (reply in 188)
188 14.92301...	192.168.0.106	192.168.0.108	ICMP	98	Echo (ping) reply id=0x137f, seq=11/2816, ttl=64 (request in 187)
206 15.91925...	192.168.0.108	192.168.0.106	ICMP	98	Echo (ping) request id=0x137f, seq=12/3072, ttl=64 (reply in 207)
207 15.92245...	192.168.0.106	192.168.0.108	ICMP	98	Echo (ping) reply id=0x137f, seq=12/3072, ttl=64 (request in 206)
208 15.92484...	192.168.0.106	192.168.0.108	ICMP	98	Echo (ping) reply id=0x137f, seq=12/3072, ttl=64
226 16.92106...	192.168.0.108	192.168.0.106	ICMP	98	Echo (ping) request id=0x137f, seq=13/3328, ttl=64 (no response found!)
233 17.68397...	192.168.0.106	192.168.0.108	ICMP	98	Echo (ping) reply id=0x137f, seq=13/3328, ttl=64 (request in 226)
238 17.92139...	192.168.0.108	192.168.0.106	ICMP	98	Echo (ping) request id=0x137f, seq=14/3584, ttl=64 (no response found!)
245 18.70638...	192.168.0.106	192.168.0.108	ICMP	98	Echo (ping) reply id=0x137f, seq=14/3584, ttl=64 (request in 238)

Como se puede observar en la captura, los paquetes que se perdieron son aquellos que aparecen con la aclaración “no response found!”.

A continuación se detallan las estadísticas del comando ping al host cliente.

```
--- 192.168.0.106 ping statistics ---
20 packets transmitted, 19 received, +4 duplicates, 5% packet loss, time 19085ms
rtt min/avg/max/mdev = 2.293/1954.875/8219.970/2644.183 ms, pipe 9
```

En ella se puede apreciar la cantidad (en porcentaje) de paquetes perdidos, gracias a lo establecido en el comando tc.

- c) Duplicación: Simule una situación de duplicación de paquetes, luego utilice la herramienta ping para verificar que efectivamente las reglas fueron correctamente definidas en el host. Detalle la configuración establecida. Realice una captura para visualizar el comportamiento y documente lo hallado.

## Trabajo Práctico #4

# Calidad de Servicio QoS

Fecha de Entrega: 13-10-2016

Medina, Ma Victoria [117024]

medina.vicc@gmail.com

```
root@viccm-C500:/home/viccm# tc qdisc replace dev wlp1s0 root netem duplicate 25%
```

```
PING 192.168.0.106 (192.168.0.106) 56(84) bytes of data.
64 bytes from 192.168.0.106: icmp_seq=1 ttl=64 time=2637 ms
64 bytes from 192.168.0.106: icmp_seq=2 ttl=64 time=1630 ms
64 bytes from 192.168.0.106: icmp_seq=3 ttl=64 time=623 ms
64 bytes from 192.168.0.106: icmp_seq=4 ttl=64 time=297 ms
64 bytes from 192.168.0.106: icmp_seq=5 ttl=64 time=28.0 ms
64 bytes from 192.168.0.106: icmp_seq=6 ttl=64 time=117 ms
64 bytes from 192.168.0.106: icmp_seq=7 ttl=64 time=209 ms
64 bytes from 192.168.0.106: icmp_seq=7 ttl=64 time=209 ms (DUP!)
64 bytes from 192.168.0.106: icmp_seq=8 ttl=64 time=6.16 ms
64 bytes from 192.168.0.106: icmp_seq=9 ttl=64 time=3.69 ms
64 bytes from 192.168.0.106: icmp_seq=10 ttl=64 time=3.30 ms
64 bytes from 192.168.0.106: icmp_seq=11 ttl=64 time=7.72 ms
64 bytes from 192.168.0.106: icmp_seq=11 ttl=64 time=7.74 ms (DUP!)
64 bytes from 192.168.0.106: icmp_seq=12 ttl=64 time=950 ms
64 bytes from 192.168.0.106: icmp_seq=13 ttl=64 time=973 ms
64 bytes from 192.168.0.106: icmp_seq=14 ttl=64 time=10.7 ms
64 bytes from 192.168.0.106: icmp_seq=15 ttl=64 time=8.65 ms
64 bytes from 192.168.0.106: icmp_seq=16 ttl=64 time=2.66 ms
64 bytes from 192.168.0.106: icmp_seq=16 ttl=64 time=4.46 ms (DUP!)
64 bytes from 192.168.0.106: icmp_seq=17 ttl=64 time=3.72 ms
64 bytes from 192.168.0.106: icmp_seq=18 ttl=64 time=4.04 ms
64 bytes from 192.168.0.106: icmp_seq=19 ttl=64 time=3.18 ms
64 bytes from 192.168.0.106: icmp_seq=20 ttl=64 time=109 ms

--- 192.168.0.106 ping statistics ---
20 packets transmitted, 20 received, +3 duplicates, 0% packet loss, time 19035ms
rtt min/avg/max/mdev = 2.664/341.416/2637.612/638.531 ms, pipe 3
```

Al realizar la captura, no se puede apreciar tan claramente como la salida por pantalla de la terminal. En la misma se puede ver los paquetes duplicados con la aclaración (DUP!) la cual nos indica que ese paquete se encuentra dos (por lo menos) veces.

- d) Re ordenamiento: Simule una situación de re ordenamiento de paquetes, donde un 25% (con una correlación del 50%) sean enviados inmediatamente mientras que el resto tenga un retraso de 10 ms. Pruebe estableciendo una sesión ssh y verifique el comportamiento a través de una captura.
- e) Piense qué aplicaciones se ven afectadas tanto por éstas (delay, pérdida, duplicación y re ordenamiento) como por otras métricas vistas en clase, y realice una comparación entre ellas, estableciendo cuáles deberían exigirse y cuáles podrían resultar más leves en relación al tipo de servicio prestado, explicando por qué en cada caso.

Es muy común ver en los diferentes noticieros cuando realizan una comunicación con un periodista que se encuentra en el exterior (por poner un caso bien notorio) que existe un tiempo muerto entre lo que se dice en el estudio hasta que el otro periodista termina de escucharlo. Cuando vemos una situación como esa, estamos en presencia delay. Es

**Trabajo Práctico #4**  
**Calidad de Servicio QoS**

**Fecha de Entrega: 13-10-2016**

Medina, Ma Victoria [117024]

medina.vicc@gmail.com

decir, el mensaje llega aunque con un retardo significativo. Nosotros, como espectadores nos resulta molesto y a veces los mismo reporteros se “pisan” a hablar.

Siguiendo con el ejemplo planteado, también podemos notar que a veces la transmisión es “mala” porque el mensaje se escucha entre cortado. En ese caso lo que se está produciendo es una pérdida de paquetes significativa (ya que en este tipo de comunicación no se establece una sesión TCP), la cual da como resultado una comunicación infructuosa.

Otro escenario a considerar es que el mensaje llegue, pero de manera desordenada o repetido. En ese caso, es casi imposible comprender la esencia del mismo.

Habiendo dicho esto, lo que se quería dejar en claro es que aquellos servicios que corren en tiempo real son los mas sensibles a este tipo de situaciones. Si en vez de situarnos en la transmisión de una conversación periodística nos ponemos en el caso de sistemas como el bypass, ocasionarían un resultado catastrófico si sucediera un error cualquiera de ellas métricas propuestas.

Como conclusión, hay ciertas situaciones donde la exigencias de estas métricas deban ser satisfechas de manera estrictas y otras, donde nos podemos dar el lujo de tolerar cierto margen de error.