UNIVERSIDAD SAN CARLOS DE GUATEMALA

CENTRO UNIVERSITARIO DE OCCIDENTE

DIVISIÓN CIENCIAS DE LA INGENIERÍA

CARRERA DE INGENIERÍA CIENCIAS Y SISTEMAS

REDES DE COMPUTADORAS 2

"**OCTAVO SEMESTRE"**

ING: JUAN FRANCISCO ROJAS

ESTUDIANTE: JOSÉ CARLOS SOBERANIS RAMÍREZ - 201730246

TRABAJO: PROYECTO 1

FECHA: 20 DE SEPTIEMBRE DE 2021

ÍNDICE

MARCO TEÓRICO

Conceptos

HTB

Es una herramienta sencilla y escalable de gestión de ancho de banda. Controla el ancho de banda para cada host o red que pasa a través de su router o firewall. Prioriza el tráfico para protocolos (Web, correo electrónico, juegos, FTP, VoIP y streaming), pero su utilización preferida es para el tráfico no deseado de sistemas de abuso de descargas (Kazaa, Emule, etc.). Posee interfaz Web para su configuración y manejo, y en casos posibles acceder de forma remota.

Protocolos de red

Un protocolo de red designa el conjunto de reglas que rigen el intercambio de información a través de una red de computadoras.

Este protocolo funciona de la siguiente forma, cuando se transfiere información de un ordenador a otro, por ejemplo mensajes de correo electrónico o cualquier otro tipo de datos esta no es transmitida de una sola vez, sino que se divide en pequeñas partes.

Ancho de banda

En computación de redes y en biotecnología, ancho de banda digital, ancho de banda de red o simplemente ancho de banda es la medida de datos y recursos de comunicación disponible o consumida expresados en bit/s o múltiplos de él como serían los Kbit/s, Mbit/s y Gigabit/s.

Disciplinas de colas

La flexibilidad y el control del control del tráfico de Linux se pueden liberar a través de la agencia de las qdiscs con clase. Recuerde que las disciplinas de cola con clase pueden tener filtros adjuntos, lo que permite que los paquetes se dirijan a clases y subcolas particulares.

Hay varios términos comunes para describir las clases directamente adjuntas a la qdisc raíz y las clases de terminal. Las clases adjuntas a la qdisc raíz se conocen como clases raíz y, más genéricamente, clases internas. Cualquier clase de terminal en una disciplina de cola en particular se conoce como una clase hoja por analogía con la estructura de árbol de las clases. Además del uso del lenguaje figurado que representa la estructura como un árbol, el lenguaje de las relaciones familiares también es bastante común.

Herramientas

tc

Es el programa de utilidad de espacio de usuario utilizado para configurar el programador de paquetes del kernel de Linux. Tc generalmente se empaqueta como parte del paquete iproute2.

crontab

El nombre cron viene del griego chronos que significa “tiempo”. En el sistema operativo Unix, cron es un administrador regular de procesos en segundo plano (demonio) que ejecuta procesos o guiones a intervalos regulares (por ejemplo, cada minuto, día, semana o mes). Los procesos que deben ejecutarse y la hora en la que deben hacerlo se especifican en el fichero crontab.

iptables

Es un programa de utilidad de espacio de usuario que permite a un administrador de sistema para configurar las tablas proporcionadas por el cortafuegos del núcleo Linux y las cadenas y reglas que almacena.

netcat

Es una herramienta de red que permite a través de intérprete de comandos y con una sintaxis sencilla abrir puertos TCP/UDP en un HOST, asociar una shell a un puerto en concreto y forzar conexiones UDP/TCP.

METODOLOGÍA

Dependencias

Centos

- yum update
- yum install epel-release
- yum install iperf3
- yum install nc

Debian

- iptables
- tc-htb
- crontab

Clientes

Configuración IP's

El default gateway para cada uno de los clientes es la dirección 10.10.10.1/24, la cual es la dirección que posee la interfaz de Debian que conecta con la red de los clientes. Las direcciones ip de los clientes son las siguientes.

- Cliente a: 10.10.10.2/24
- Cliente b: 10.10.10.3/24
- Cliente c: 10.10.10.4/24

Configuración de iptables

Una vez asignadas las direcciones IP así como haber configurado su default gateway, procederemos a liberar de todas las reglas predeterminadas en centos para el control de paquetes debido a que el filtrado de paquetes se realizará en debian, esto se realizará ejecutando

iptables -F

Simulador de ISP

Configuración IP

Para la máquina de centos en la que ocasionalmente realizaremos pruebas, asignaremos la dirección 10.10.8.1/24, la cual tendrá conexión con una interfaz de debian que se encuentra bajo la misma red, teniendo también que eliminar todas las reglas predeterminadas que centos posee con

iptables -F

Servidor (Debian) - Configuraciones

Control de ancho de banda

Para el control de ancho de banda se utilizó tc-htb, con ello podemos dividir el flujo de los paquetes por caminos distintos, a los cuales serán asignados según un filtro, inicialmente deberemos crear la disciplina que tendrá nuestro nodo raíz, utilizando:

```
tc qdisc add dev $DEV root handle 1: htb default 10
#DEV sería nuestra interfaz
#handle sería el identificador
#default hace referencia a la 'clase' por la que deberán enviarse los paquetes de manera predeterminada
```

Después procederemos a crear las clases necesarias, en nuestro caso 3, para 3 clientes

```
tc class add dev $DEV parent 1:1 classid 1:10 htb rate 1Kbit
tc class add dev $DEV parent 1:1 classid 1:11 htb rate 1Kbit
tc class add dev $DEV parent 1:1 classid 1:12 htb rate 1Kbit
tc class add dev $DEV parent 1:1 classid 1:13 htb rate 1Kbit
#classid será el identificador de nuestra clase
#rate será el ancho de banda mínimo que tendrá (si no se coloca ceil la cantidad que coloquemos será la mínima y máxima)
```

Control de protocolos

Para controlar el flujo de paquetes utilizando los diferentes tipos de protocolos se utilizó iptables, esta herramienta nos ayudó a limitar el flujo a través de la solicitud de ciertos puertos, así como al uso de cierto tipo de protocolos.

icmp

```
iptables -I FORWARD 1 -p icmp -m mac --mac-source $MAC -j ACCEPT
iptables -I FORWARD 1 -p icmp -m state --state RELATED,ESTABLISHED -j ACCEPT

#-I FORWARD 1 indica que será una regla para los paquetes que 'fluyen' a través de Debian, pero que no salen de Debian o que entren a Debian, el 1 que la regla será insertada en la posición 1

#-p icmp indica el protocolo
```

```
#-m mac --mac-source carga un módulo de iptables, el módulo es mac y 'la función' que utilizaremos es la --mac-source, lo cual nos permite que una mac pueda enviar paquetes icmp a través de Debian.

#-m state --state ESTABLISHED,RELATED nos permite que sólo puedan pasar paquetes de Debian que tengan una conexión o la hayan tenido, así evitando que accedan desde el ISP a los clientes, si este no solicitó nada anteriormente.
```

udp/tcp

```
iptables -I FORWARD 1 -p $PROTO -m mac --mac-source $MAC -m $PROTO --dport $PUERTO-INICIO:$PUERTO-FINAL -j ACCEPT

iptables -I FORWARD 1 -p $PROTO -m state --state RELATED,ESTABLISHED -m $PROTO --sport $PUERTO-INICIO:$PUERTO-FINAL -j ACCEPT

#--dport hace referencia al puerto destino, es decir, el puerto hacia el que el cliente envía una solicitud, de esta forma logramos que la mac que necesitemos pueda enviar paquetes udp/tcp a cierto lugar con cierto puerto.

#--sport hace referencia al puerto origen, es decir, el puerto que responde a una solicitud que envió el cliente.

#Al igual que con el protocolo icmp, desde el ISP hacia el cliente, sólo aceptaremos los paquetes cuya conexión exista o haya existido, así evitaremos que se envíen paquetes desde el ISP que el cliente no haya solicitado

#Estas reglas son para un rango de puertos, para realizar un único puerto, simplemente colocaremos después de los parámetros sport/dport <puerto>
```

Control por horarios

Para el control por horarios, se utilizó la herramienta crontab, esta ejecuta ciertas tareas que nosotros programemos a la hora que nosotros deseemos, bajo esta premisa, configuramos que a partir de ciertas horas se agregan ciertas reglas a iptables y que al finalizar el tiempo se eliminen. Se hará escribiendo en el archivo de tareas /var/spool/cron/crontabs/root, y tendrá la siguiente sintaxis:

```
$MINUTO $HORA * * * $COMANDO-A-EJECUTAR
```

Servidor (Debian) - Automatización

Archivos .conf

Estos serán los archivos de entrada para el sistema de control.

MACS.conf

El archivo MACS tendrá dentro de él las direcciones MAC de las diferentes máquinas que formarán parte del control de ancho de banda y puertos. Ej:

```
MAC1=52:54:00:6C:ab:73
MAC2=52:54:00:a6:52:01
MAC3=52:54:00:ab:c3:7b
```

usuario_bw.conf

Este archivo tendrá dentro las reglas para limitar el ancho de banda para cada uno de los clientes por mac, la dirección MAC será referenciada desde el archivo MACS.conf, su formato es: <MAC>,<% bajada>,<% subida>,<hora-inicio>,<hora-finalización>

```
MAC1,30,30,19:00,20:00
MAC2,30,30,19:00,20:00
MAC3,30,30,19:00,20:00
MAC1,30,30,20:14,21:00
MAC2,30,30,20:14,21:00
MAC3,30,30,20:14,21:00
MAC1,20,20,21:55,23:00
MAC2,20,20,21:55,23:00
MAC3,20,20,21:55,23:00
```

usuario_proto.conf

Este archivo tendrá dentro las reglas para limitar el acceso a puertos para cada uno de los clientes por mac, la dirección MAC será referenciada desde el archivo MACS.conf, su formato es: <MAC>,<protocolo>[,<puerto(s)>],<hora-inicio>,<hora-finalización>

```
MAC1,icmp,20:55,21:20
```

```
MAC2,icmp,20:55,21:20
MAC3,icmp,20:55,21:20
MAC1,tcp,6100,20:55,21:20
MAC2,tcp,6101,20:55,21:20
MAC3,tcp,6102,20:55,21:20
MAC1,udp,6200,20:55,21:20
MAC2,udp,6201,20:55,21:20
MAC3,udp,6202,20:55,21:20
```

modo.conf

Este archivo definirá el modo en el que estará trabajando el ancho de banda, estará definido por la siguiente estructura:

modalidad = <número que identifica al tipo de modalidad>

Los tipos pueden ser:

- 1 = (estricto) de este modo el ancho de banda será únicamente el especificado, no más.
- 2 = (dinámico) de este modo el ancho de banda será como mínimo lo especificado o más.

enlace.conf

Este archivo definirá la velocidad que tendrán de subida y de bajada, su estructura estará definida por:

down = <cantidad en Mbps>
up = <cantidad en Mbps>

Archivos .sh

Los archivos .sh son los ejecutables encargados de realizar las configuraciones para que se ejecuten los cambios necesarios a las horas correspondientes.

limpieza.sh

Al ejecutar este archivo todas las configuraciones realizadas por ciclos anteriores serán eliminadas, dejando las configuraciones por defecto.

conf-inicial.sh

- Crea el nodo raíz, así como todas las clases que derivarán de él con un ancho de banda de 1Kbit (dado que habrá ancho de banda limitado, pero no se puede configurar 0)
- Crea los filtros por mac para el ancho de banda y de esta forma decidir a través de qué clase se realizará la conexión.
- Cambia las políticas de iptables, dejando de manera predeterminada dropeando todos los mensajes.

exec.sh

- Calcula el ancho de banda total que manejará la interfaz de debian conectada a los clientes
- Determina el modo en el que se ejecutará el ancho de banda, si será estricto o dinámico
- Llamará a los .sh iptables-command.sh y insertar-crontab.sh según las reglas leídas en el archivo usuario_proto.conf y usuario_bw.conf

iptables-command.sh

Este escribirá en el archivo de tareas programadas de crontab los comandos que deben ejecutarse para realizar los cambios de ancho de banda en la red.

insertar-crontab.sh

Este escribirá en el archivo de tareas programadas de crontab los comandos que deben ejecutarse para realizar los cambios en los accesos a los puertos y protocolos.

RESULTADOS

Limitando ancho de banda en cliente a y cambiando durante la descarga de un archivo de prueba

Inicialmente

```
class htb 1:11 root prio 0 quantum 12800 rate 1024Kbit ceil 1024Kbit linklayer ethernet burst 1599b/1 mpu 0b cbu
rst 1599b/1 mpu 0b level 0
 Sent 101368722 bytes 66984 pkt (dropped 0, overlimits 57321 requeues 0)
 backlog 523844b 314p requeues 0
 lended: 57363 borrowed: 0 giants: 0
 tokens: -183742 ctokens: -183742
```

Cliente - a

```
[root@localhost ~]# wget ipv4.download.thinkbroadband.com/1GB.zip
--2021-09-19 16:46:09--  http://ipv4.download.thinkbroadband.com/1GB.zip
Resolviendo ipv4.download.thinkbroadband.com (ipv4.download.thinkbroadband.com)... 80.249.99.148
Conectando con ipv4.download.thinkbroadband.com (ipv4.download.thinkbroadband.com)[80.249.99.148]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 1073741824 (1.0G) [application/zip]
Grabando a: “1GB.zip"

10% [========>                                                                ] 112,434,023  120KB/s  T.E. 74m 42s
```

Cliente sin filtrar utilizando el ancho de banda restante

```
[root@localhost ~]# wget ipv4.download.thinkbroadband.com/1GB.zip
--2021-09-19 16:53:32--  http://ipv4.download.thinkbroadband.com/1GB.zip
Resolviendo ipv4.download.thinkbroadband.com (ipv4.download.thinkbroadband.com)... 80.249.99.148
Conectando con ipv4.download.thinkbroadband.com (ipv4.download.thinkbroadband.com)[80.249.99.148]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 1073741824 (1.0G) [application/zip]
Grabando a: “1GB.zip"

14% [===========>                                                             ] 153,623,831 1.18MB/s  T.E. 12m 33s
```

Cambiando el ancho de banda para el cliente a

```
root@debian-fxhnd:/home/fxhnd/Documentos/8vo-Redes2-Proyecto1/ejecutables# tc -s -d class show dev ens8
class htb 1:11 root prio 0 quantum 38375 rate 3070Kbit ceil 3070Kbit linklayer ethernet burst 1599b/1 mpu 0b cbu
rst 1599b/1 mpu 0b level 0
 Sent 130187712 bytes 86019 pkt (dropped 0, overlimits 75336 requeues 0)
 backlog 339136b 211p requeues 0
 lended: 75386 borrowed: 0 giants: 0
 tokens: -61427 ctokens: -61427
```

Cliente - a

```
[root@localhost ~]# wget ipv4.download.thinkbroadband.com/1GB.zip
--2021-09-19 16:46:09--  http://ipv4.download.thinkbroadband.com/1GB.zip
Resolviendo ipv4.download.thinkbroadband.com (ipv4.download.thinkbroadband.com)... 80.249.99.148
Conectando con ipv4.download.thinkbroadband.com (ipv4.download.thinkbroadband.com)[80.249.99.148]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 1073741824 (1.0G) [application/zip]
Grabando a: “1GB.zip"

12% [=========>                                                                ] 134,850,511  358KB/s  T.E. 74m 52s
```

Cliente sin filtrar utilizando el ancho de banda restante

```
[root@localhost ~]# wget ipv4.download.thinkbroadband.com/1GB.zip
--2021-09-19 16:53:32--  http://ipv4.download.thinkbroadband.com/1GB.zip
Resolviendo ipv4.download.thinkbroadband.com (ipv4.download.thinkbroadband.com)... 80.249.99.148
Conectando con ipv4.download.thinkbroadband.com (ipv4.download.thinkbroadband.com)[80.249.99.148]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 1073741824 (1.0G) [application/zip]
Grabando a: “1GB.zip"

23% [===================>                                                    ] 254,668,167  975KB/s  T.E. 11m 34s
```

Probando ancho de banda limitado para los diferentes clientes (simultáneamente)

```
root@debian-fxhnd:/home/fxhnd/Documentos/8vo-Redes2-Proyecto1/ejecutables# tc -s -d class show dev ens8
class htb 1:11 root prio 0 quantum 38375 rate 3070Kbit ceil 3070Kbit linklayer ethernet burst 1599b/1 mpu 0b cbu
rst 1599b/1 mpu 0b level 0
 Sent 182042302 bytes 120270 pkt (dropped 0, overlimits 105383 requeues 0)
 backlog 485994b 268p requeues 0
 lended: 105591 borrowed: 0 giants: 0
 tokens: -61354 ctokens: -61354

class htb 1:10 root prio 0 quantum 1000 rate 1Kbit ceil 1Kbit linklayer ethernet burst 1600b/1 mpu 0b cburst 160
0b/1 mpu 0b level 0
 Sent 2030 bytes 47 pkt (dropped 0, overlimits 0 requeues 0)
 backlog 0b 0p requeues 0
 lended: 47 borrowed: 0 giants: 0
 tokens: 194750000 ctokens: 194750000

class htb 1:13 root prio 0 quantum 12800 rate 1024Kbit ceil 1024Kbit linklayer ethernet burst 1599b/1 mpu 0b cbu
rst 1599b/1 mpu 0b level 0
 Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
 backlog 0b 0p requeues 0
 lended: 0 borrowed: 0 giants: 0
 tokens: 200000000 ctokens: 200000000

class htb 1:12 root prio 0 quantum 25600 rate 2048Kbit ceil 2048Kbit linklayer ethernet burst 1599b/1 mpu 0b cbu
rst 1599b/1 mpu 0b level 0
 Sent 1548 bytes 14 pkt (dropped 0, overlimits 0 requeues 0)
 backlog 0b 0p requeues 0
 lended: 14 borrowed: 0 giants: 0
 tokens: 92162 ctokens: 92162
```

Cliente - a

```
[root@localhost ~]# wget ipv4.download.thinkbroadband.com/1GB.zip
--2021-09-19 16:46:09--  http://ipv4.download.thinkbroadband.com/1GB.zip
Resolviendo ipv4.download.thinkbroadband.com (ipv4.download.thinkbroadband.com)... 80.249.99.148
Conectando con ipv4.download.thinkbroadband.com (ipv4.download.thinkbroadband.com)[80.249.99.148]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 1073741824 (1.0G) [application/zip]
Grabando a: “1GB.zip"

14% [============>                                                    ] 160,582,919  357KB/s  T.E. 67m 51s
14% [============>                                                    ] 160,656,767  358KB/s  T.E. 67m 51s
14% [============>                                                    ] 160,732,063  358KB/s  T.E. 67m 46s
14% [============>                                                    ] 160,805,911  358KB/s  T.E. 67m 46s
24% [====================>                                            ] 259,114,975  359KB/s  T.E. 51m 36s
```

Cliente - b

```
[root@localhost ~]# wget ipv4.download.thinkbroadband.com/1GB.zip
--2021-09-19 17:00:10--  http://ipv4.download.thinkbroadband.com/1GB.zip
Resolviendo ipv4.download.thinkbroadband.com (ipv4.download.thinkbroadband.com)... 80.249.99.148
Conectando con ipv4.download.thinkbroadband.com (ipv4.download.thinkbroadband.com)[80.249.99.148]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 1073741824 (1.0G) [application/zip]
Grabando a: “1GB.zip"

 3% [==>                                                              ] 37,605,727   239KB/s  T.E. 70m 39s
```

Cliente - c

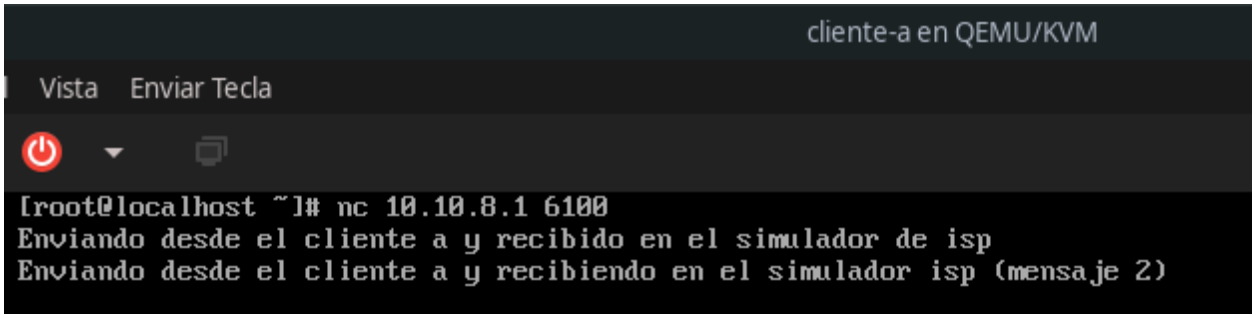
```
[root@localhost ~]# wget ipv4.download.thinkbroadband.com/1GB.zip
--2021-09-19 17:03:41--  http://ipv4.download.thinkbroadband.com/1GB.zip
Resolviendo ipv4.download.thinkbroadband.com (ipv4.download.thinkbroadband.com)... 80.249.99.148
Conectando con ipv4.download.thinkbroadband.com (ipv4.download.thinkbroadband.com)[80.249.99.148]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 1073741824 (1.0G) [application/zip]
Grabando a: “1GB.zip"

 0% [                                                                ] 920,647      120KB/s  T.E. 2h 26m
```

Admitiendo envío de paquetes a través de Debian para el protocolo udp puerto 6200 desde el cliente a

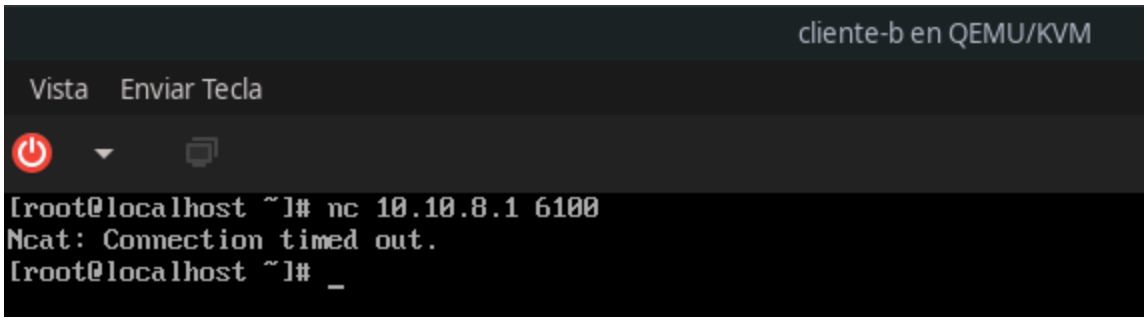
Configuraciones debian

```
root@debian-fxhnd:~# iptables -S
-P INPUT DROP
-P FORWARD DROP
-P OUTPUT DROP
-A FORWARD -p tcp -m state --state RELATED,ESTABLISHED -m tcp --sport 6100 -j ACCEPT
-A FORWARD -p tcp -m mac --mac-source 52:54:00:6c:ab:73 -m tcp --dport 6100 -j ACCEPT
root@debian-fxhnd:~#
```

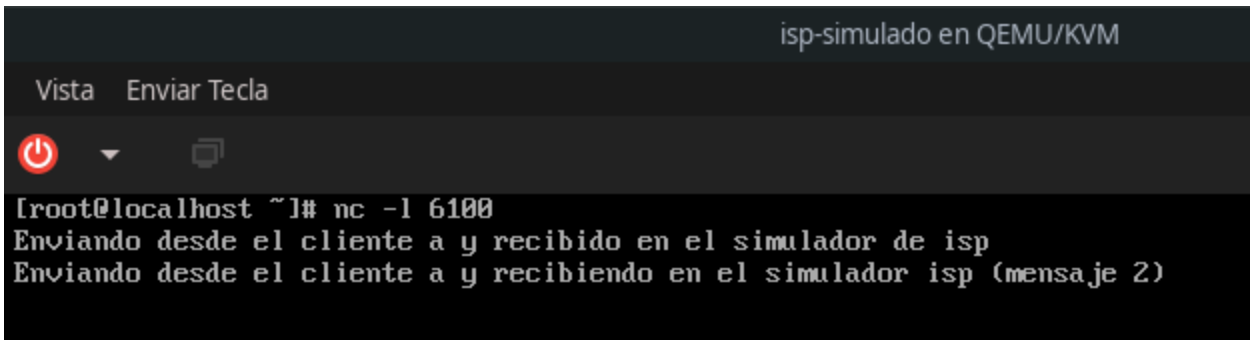
Cliente a



Cliente b



ISP simullado

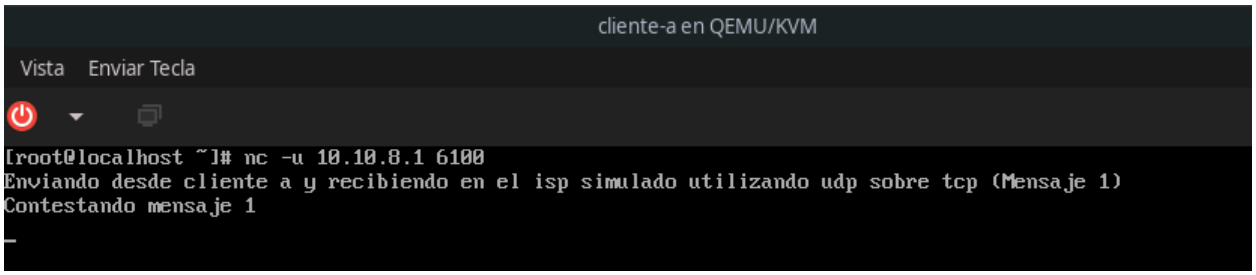


Admitiendo envío de paquetes a través de Debian para el protocolo tcp puerto 6100 desde el cliente a

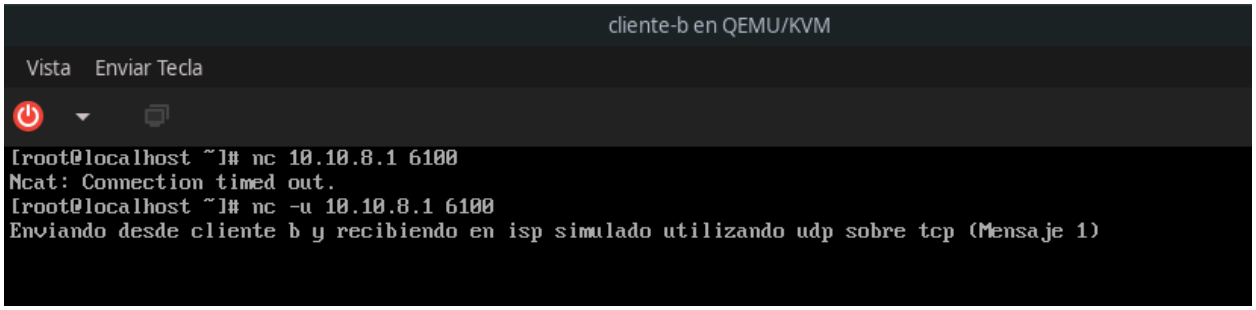
Configuraciones debian

```
root@debian-fxhnd:~# iptables -S
-P INPUT DROP
-P FORWARD DROP
-P OUTPUT DROP
-A FORWARD -p udp -m state --state RELATED,ESTABLISHED -m udp --sport 6100 -j ACCEPT
-A FORWARD -p udp -m mac --mac-source 52:54:00:6c:ab:73 -m udp --dport 6100 -j ACCEPT
root@debian-fxhnd:~#
```

Cliente a



Cliente b



ISP simulado

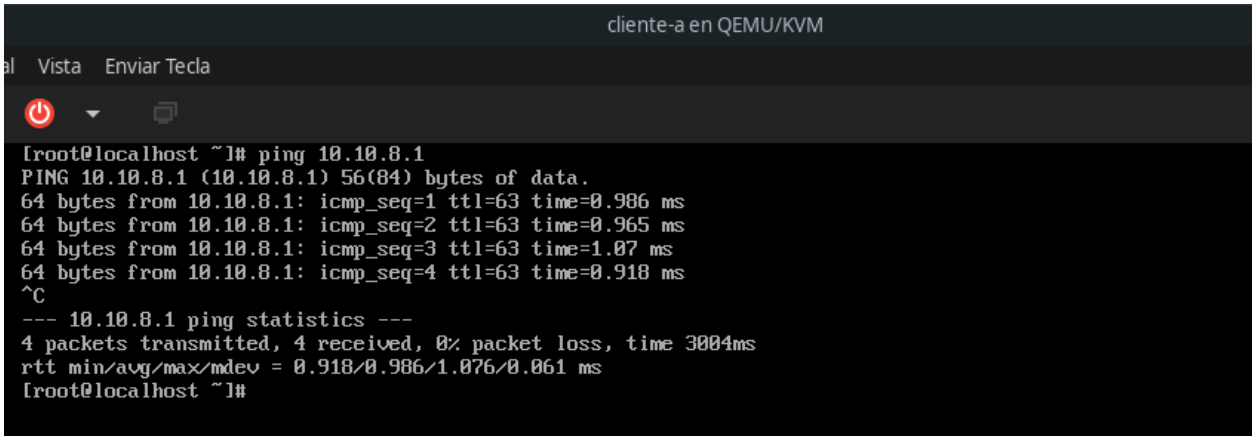
```
[root@localhost ~]# nc -ul 6100
Enviando desde cliente a y recibiendo en el isp simulado utilizando udp sobre tcp (Mensaje 1)
Contestando mensaje 1
```

Admitiendo envío de paquetes a través de Debian para el protocolo icmp

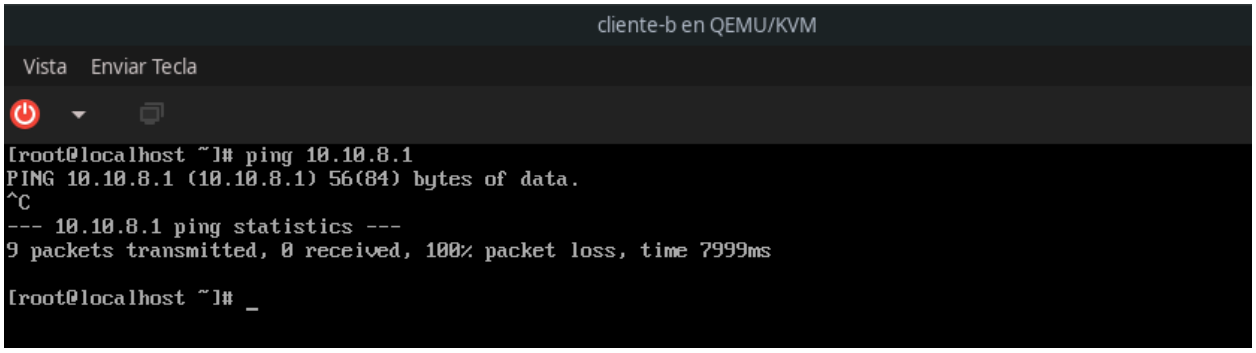
Configuraciones debian

```
root@debian-fxhnd:~# iptables -S
-P INPUT DROP
-P FORWARD DROP
-P OUTPUT DROP
-A FORWARD -p icmp -m mac --mac-source 52:54:00:6c:ab:73 -j ACCEPT
-A FORWARD -p icmp -m state --state RELATED,ESTABLISHED -j ACCEPT
root@debian-fxhnd:~#
```

Cliente a



Cliente b



REFERENCIAS

- Borque, J. L. S. (2017, 1 junio). *Tutorial IPTABLES II – Entendiendo el flujo de paquetes*. myTCPIP blog by @sanchezborque. https://mytcpip.com/tutorial-iptables-ii-entendiendo-el-flujo-de-paquetes/
- *Classful Queuing Disciplines (qdiscs)*. (s. f.). HOWTO. Recuperado 20 de septiembre de 2021, de https://tldp-org.translate.goog/HOWTO/Traffic-Control-HOWTO/classful-qdiscs.html?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=ajax,sc,elem#qc-htb
- Dumitru, D. (2020, 28 mayo). *Example5: Traffic Prioritizing with HTB and MAC filtering*. OpenWrt Wiki. https://openwrt.org/docs/guide-user/network/traffic-shaping/packet.scheduler.example5
- *How to limit speed for every device per MAC address in the gateway via Linux command «tc»?* (2017, 21 febrero). Server Fault. https://serverfault.com/questions/833862/how-to-limit-speed-for-every-device-per-mac-address-in-the-gateway-via-linux-com
- *Limitar el ancho de banda para un programa o una interfaz (Pagina 1) / Wireless y redes en linux. / Foro Wifi-libre.com*. (s. f.). Wifi-libre. Recuperado 20 de septiembre de 2021, de https://www.wifi-libre.com/topic-32-limitar-el-ancho-de-banda-para-un-programa-o-una-interfaz.html
- Mauro, D. (2018, 28 junio). *Linux* [Foro en taringa]. Taringa! https://www.taringa.net/+linux/squid-limitar-ancho-de-banda_i0qn6

- *¿Qué es Netcat y cómo funciona?* (2020, 2 octubre). IONOS Digitalguide. https://www.ionos.es/digitalguide/servidores/herramientas/netcat/
- Sanders, L. (2005, 1 enero). *[LARTC] Marking packets by mac addr using tc filter u32 match?* Narkive. https://boerde.lists.lartc.narkive.com/GhiJEp5m/lartc-marking-packets-by-mac-addr-using-tc-filter-u32-match
- *Shaping Linux Traffic with tc*. (2020, 16 septiembre). Octetz. https://octetz.com/docs/2020/2020-09-16-tc/
- *tc-htb(8) - Linux manual page*. (2002, 10 enero). Man7. https://man7.org/linux/man-pages/man8/tc-htb.8.html
- TrafficControl - Debian Wiki. (2019, 9 marzo). En *WikiDebian*. https://wiki.debian.org/TrafficControl
- Vargas Castro, L. (2018, 18 octubre). *Distribución del ancho de banda utilizando HTB e IPTables - Alcance Libre*. AlcanceLibre. https://www.alcancelibre.org/staticpages/index.php/distribucion-ancho-banda-htb-iptables