

# Tecnologies Avançades d'Internet

## HOWTO Pràctica 2: *Infraestructura iproute2*

### 1 Permetre l'emascament o no segons la subxarxa d'origen

Per a fer la traducció d'adreces de les nostres subxarxes d'entrada definirem tantes regles de iptables com **subxarxes** tinguem.

Noteu que el gateway ha de poder utilitzar, per a la traducció d'adreces, tant la IP associada a la seva interfície **eth0** com la de l'interfície **eth1**.

### 2 Configuració de les taules de routing

#### 2.1 Comprovació de les taules actuals

Al vincular adreces IP a les interfícies del sistema, el mateix kernel afegeix les entrades corresponents a les taules de routing. Podeu comprovar el contingut d'aquesta taula amb la comanda “`ip r s`”.

Realment, el kernel del sistema operatiu utilitza més taules per poder enrutar de forma correcta tots els paquets cap a les seves corresponents destinacions. Amb la comanda “`ip rule show`” hauria d'aparèixer una taula com la següent:

```
0: from all lookup local
32766: from all lookup main
32767: from all lookup default
```

Aquestes són les taules que realment utilitza el kernel del sistema quan necessita prendre una decisió d'enrutament. Per mostrar les diferents regles que contenen cadascuna de les taules podem utilitzar la comanda “`ip route show table XXX`” on “XXX” és el nom de la taula que volem veure. A la taula “local” hi ha les entrades per les direccions de broadcast, les direccions locals, etc. A “main” hi trobarem les regles que ens permeten accedir a les xarxes locals connectades a les interfícies. Per afegir o eliminar alguna de les entrades que trobem a aquestes taules heu d'utilitzar la comanda:

```
ip route [opcions]
```

## 2.2 Configurant un sol gateway de sortida

Abans de configurar el balanceig de càrrega hauríem de provar tot el que s'ha fet fins ara. Abans de fer-ho, però, haurem d'afegir una entrada per defecte a la taula de routing per poder accedir a Internet. Podem utilitzar qualsevol dels dos gateways que tenim configurats a les dues xarxes de sortida. Per a fer-ho heu d'utilitzar la comanda:

```
ip route [opcions]
```

D'aquesta manera, s'afegeix una nova regla a la taula “main”.

## 2.3 Configuració del balanceig de càrrega

Per afegir la regla de balanceig en la taula d'encaminament, utilitzareu la comanda:

```
ip route [opcions]
```

Vegeu el capítol 4.2.2 de [3].

Per canviar aquest balanceig de forma dinàmica (sense haver d'eliminar la regla a la taula de routing i tornar-la a crear) podem utilitzar la comanda:

```
ip route change default ....
```

Per comprovar que el balanceig de càrrega funciona correctament podem obrir un parell de consoles que ens mostrin el trànsit que passa per cadascuna de les dues interfícies de sortida utilitzant les següents comandes o bé el *wireshark*:

```
# tcpdump -i eth0 -vv
# tcpdump -i eth1 -vv
```

Si fem varis pings (“`ping XXX &> /dev/null &`”) des de qualsevol workstation cap a una màquina que estigui a internet, podrem veure com les dues interfícies s'utilitzen amb, aproximadament, el percentatge d'ús que els hi haguem definit.

### 3 Limitació de l'ample de banda utilitzat per cada subxarxa

#### Limitant l'ample de banda d'una sola interfície

Abans de limitar l'ample de banda per les diferents subxarxes que tinguem definides, provarem de configurar la limitació per cada una de les interfícies de sortida del gateway. Una de les polítiques de cua que podem utilitzar per a limitar l'ample de banda d'una interfície és la *tb*f (*token bucket filter*).

Si li associem aquesta política de cues a les interfícies del gateway, abans que els paquets surtin per la interfície passaran per aquesta cua. Així, a partir dels paràmetres que li configurem a la *tb*f, podrem limitar la velocitat de transferència màxima.

Per afegir aquest tipus de cua a la interfície *eth0* podem utilitzar la següent comanda:

```
# tc qdisc add dev eth0 root tbf rate 10kbit latency 50ms burst 1540
```

On els paràmetres:

**rate** ens permet definir una taxa màxima de transferència.

**latency** és el retard.

**burst** és el tamany del buffer que utilitza la cua.

Es pot intentar variar els paràmetres de *latency* i *burst* perquè el paràmetre *rate* sigui realment l'ample de banda desitjat. De totes maneres, i per no complicar-nos massa, per aquesta pràctica no fa falta fer-ho.

Per provar que la configuració funciona correctament, des de dues consoles diferents podeu executar una de les següents comandes en cada consola:

```
watch -n 1 tc -s qdisc show dev eth0
watch -n 1 tc -s qdisc show dev eth1
```

Haureu de veure la cua que heu creat. Des d'una altra consola genereu tràfic. En la consola del `watch` veureu com els datagrames s'encuen en una de les dues cues.

Per a eliminar aquesta configuració podeu executar la comanda:

```
tc qdisc del dev eth0 root tbf
```

## 3.1 Limitació de l'ample de banda segons la subxarxa d'origen

### 3.1.1 Crear la cua de tipus PRIO

Abans de configurar el router perquè ens limiti el bandwidth segons la subxarxa d'origen, haurem d'eliminar la cua que hem configurat a l'apartat anterior.

A continuació, per a cada interfície, haurem de crear una cua amb tantes bandes com subxarxes vulguem configurar. Hem de tenir en compte que, a més de la banda que necessitem per cada subxarxa, n'hem de crear una de més pels paquets que provinguin del mateix router (per no limitar-li el seu propi ample de banda).

Una cua de tipus `PRIO`, prioritats, és una cua *classfull* i per tant ens permet associar bandes (subcues) a aquesta cua `PRIO`.

Si, per exemple, tinguéssim 4 subxarxes d'entrada amb els identificadors:

```
192.168.2.0/26, 192.168.2.64/26, 192.168.2.128/26 i 192.168.2.192/26
```

La comanda que hauríem d'executar per crear una cua `PRIO` amb 4 bandes seria:

```
# tc qdisc add dev eth0 root handle 1: prio bands 5
```

A continuació us indiquem què volen dir alguns dels paràmetres de la comanda anterior.

**Paràmetre dev eth0:** Estem creant una cua associada a la interfície `eth0`.

**Paràmetre prio:** Ens indica que estem creant una cua de tipus PRIO.

**root handle 1:** Indiquem que la cua tindrà subcues format un esquema jeràrquic. L'arrel d'aquesta jerarquia està identificada amb l'identificador `handle 1:.`

**Paràmetre bands:** Ens indica que la cua PRIO que estem creant tindrà 5 bandes. En aquest cas 4 bandes per les subxarxes + 1 banda pels paquets del router.

### 3.1.2 Configurar les bandes de la cua PRIO

La primera banda la definirem amb un paràmetre “rate” de “10000kbit”. Aquesta banda serà la que utilitzarem per encuar els paquets del propi router. D’aquesta forma podrà aprofitar el màxim ample de banda que li proporcioni la xarxa.

Les altres bandes les definirem, per exemple, perquè la primera subxarxa tingui un “rate” de 10kbit, la segona 20kbit, etc...

**Les comandes per a definir l’ample de banda per a cada "banda" que heu creat les trobareu al capítol 9.5.3 de [3].**

Per poder canviar el paràmetre “rate” d’alguna d’aquestes bandes sense haver-les d’eliminar i tornar-les a crear utilitzarem la comanda:

```
tc qdisc change dev IF parent 1:X handle X0: tbf rate Ykbit
latency 50ms burst 1540
```

A on:

**IF** Serà la interfície de sortida on haguem configurat la cua.

**X** És el número de banda (o subcua).

**Y** És el nou rate.

### 3.1.3 Marcatge de paquet amb el `real`

El següent pas consisteix en forçar que cada datagrama generat per les workstations es dirigeixi a la banda (subcua) que li toqui, segons la seva subxarxa d’origen. Per a fer-ho **marcarem** el paquet associant-li una marca segons la seva subxarxa d’origen.

Quan un datagrama entra per alguna de les interfícies del sistema, es crea una estructura en memòria, associada al datagrama. Aquesta estructura existeix fins que el datagrama surt per la interfície de xarxa per arribar al seu destí. En aquesta estructura podem posar-hi una marca per poder identificar el datagrama posteriorment.

És necessari que marquem els paquets ja que al emmascarar les adreces, quan el paquet surt per alguna de les interfícies `eth0` o `eth1` ja no té la seva adreça IP d’origen, sinó que li hem canviat per una IP del gateway vàlida, amb la que ens

puguem connectar a internet. Per tant deixem de saber de quina subxarxa d'origen prové el datagrama.

Així doncs per cada subxarxa que tinguem definida haurem de posar una marca diferent.

**Per al marcatge utilitzarem el camp "realm" de la taula d'encaminament. Vegeu el capítol 12.2 de [3].**

El paràmetre "realm" és la marca que deixarem en el paquet.

### 3.1.4 Classificar (filtrar) els paquets segons la marca (realm)

Per dirigir el paquet a la banda (subcua) corresponent haurem de comprovar quina marca té i, segons aquesta, enviar-lo cap a una banda o una altra.

Per exemple els datagrames que provenguin de la subxarxa:

- 192.168.2.0/26 tenen la marca 2 i els hem d'enviar cap a la banda **1:2** (o, el que és el mateix, banda **20:**).
- 192.168.2.32/26 tenen la marca 3 i han d'anar a la banda **1:3** (o, el que és el mateix, banda **30:**).
- ...

Per encuar els datagrames a la banda que toca en funció de la marca del realm hem de definir uns **filtres**. Utilitzarem la comanda:

**tc filter [paràmetres].**

**Important:** Quan utilitzeu la comanda `tc filter`, per indicar a quina banda associeu el filtre, utilitzeu el paràmetre **flowid**, no pas **classid**.

**Per a més informació sobre aquesta comanda vegeu man tc i el capítol 12.2 de [3].**

Per comprovar que tot plegat funciona correctament podem obrir dues consoles i executar-hi una d'aquestes comandes en cada consola:

```
watch -n 1 tc -s qdisc show dev eth0
watch -n 1 tc -s qdisc show dev eth1
```

Amb les comandes anteriors hauríem de veure dues execucions semblants a:

```
Every 1s: tc -s qdisc show dev eth0

qdisc tbf 50: rate 40Kbit burst 1539b lat 59.6ms Sent 0 bytes 0 pkts
(dropped 0, overlimits 0)

qdisc tbf 40: rate 30Kbit burst 1539b lat 60.4ms Sent 0 bytes 0 pkts
(dropped 0, overlimits 0)

qdisc tbf 30: rate 20Kbit burst 1539b lat 57.2ms Sent 0 bytes 0 pkts
(dropped 0, overlimits 0)

qdisc tbf 20: rate 10Kbit burst 1539b lat 57.2ms Sent 0 bytes 0 pkts
(dropped 0, overlimits 0)

qdisc tbf 10: rate 10000Kbit burst 1538b lat 61.0ms Sent 0 bytes 0 pkts
(dropped 0, overlimits 0)

qdisc prio 1: bands 5 priomap 1 2 2 2 1 2 0 0 1 1 1 1 1 1 1 1
Sent 126 bytes 3 pkts (dropped 0, overlimits 0)
```

D'aquesta execució podem veure que cada banda de la cua PRIO és una `tbf`; El número de datagrames que s'envien des d'aquesta banda i el nombre de datagrames de la banda que es descarten.

Si des d'una workstation ens connectem o fem pings a alguna màquina de l'exterior, veurem com els datagrames que genera la workstation s'encuen a la banda que toca. Veurem que els datagrames d'aquesta banda s'aniran incrementant.

Assegureu-vos que, depenent de la workstation des d'on estiguem generant tràfic, tots els datagrames es redirigeixen correctament a les seves respectives bandes.



## 4 Treure la limitació de l'ample de banda del router

Ja només ens queda fer que els datagrames provinents del propi router utilitzin la banda que no té limitació d'ample de banda. Vegeu la secció 9.2.1 de [3].

Tal i com tenim la configuració fins a aquest punt podeu comprovar que els datagrames del router se situen a alguna de les bandes que hem definit.

Per defecte, la banda on van a parar els datagrames del router ve determinada pel paràmetre *priomap* de la comanda que heu utilitzat per a crear la cua de tipus PRIO.

El paràmetre *priomap* utilitza el valor del camp TOS (*Type Of Service*) del datagrama, per enviar el datagrama a una banda o una altra. **Vegeu la secció 9.2.1.1 de [3].**

Això es fa així per poder optimitzar el màxim el rendiment de la nostra xarxa.

Segons el tipus de servei que especifica una aplicació (TOS), el datagrama es pot redirigir a un tipus de banda o altre per maximitzar-ne el rendiment.

En el nostre cas, no ho tindrem en compte i farem que **tots els datagrames no marcats amb un *realm*** vagin, per defecte, a la banda on no tinguem limitat l'ample de banda.

El *priomap* d'una interfície el podem canviar amb la comanda:

```
tc qdisc change dev eth0 root handle 1: prio bands 5
priomap X X X X X X X X X X X X X X X X
```

On X l'heu de canviar pel número de banda que no té limitació d'ample de banda. Les bandes s'identifiquen des de la 0 fins bands-1.

Amb tot això, la pràctica ja funcionaria correctament.

## 5 Comandes per eliminar els filtres i les cues definides a les interfícies

**Eliminar tots els filtres i les cues definides** Si volem eliminar tots els filtres i les cues definides en una interfície podem utilitzar la comanda:

```
tc qdisc del dev ethX root handle 1:
```

**Eliminar el marcatge de paquets** Per eliminar el marcatge de paquets podem utilitzar la comanda:

```
ip route del 192.168.2.X/26 dev eth2
```

## Referències

- [1] **IP command reference.** URL: <http://linux-ip.net/gl/ip-cref/ip-cref.html>. Manual de la comanda ip.
- [2] **Manual d'iptables.** Manual del sistema de iptables accessible a partir de la comanda “man 8 iptables”.
- [3] **Linux Advanced Routing & Traffic Control.** URL: <http://www.lartc.org>. HOWTO de Linux Advanced Routing & Traffic Control utilitzant l'infraestructura de iproute2.
- [4] **VMWare homepage.** URL: <http://www.vmware.com>. Pàgina web oficial de l'empresa vmware.