

SHAPER

Pierpaolo Palazzoli, Brescia, Italy

Fabio Mostarda, Brescia, Italy

Claudia Ghelfi, Brescia, Italy

INTRODUZIONE

Lo Shaper è uno strumento che consente di sagomare il profilo di traffico su di una rete. Può essere impiegato per applicare filtri più o meno restrittivi all'intero flusso, oppure per differenziare traffici di classi diverse, assegnando ad ognuno caratteristiche specifiche.

In questo particolare caso si prende in considerazione lo strumento Shaper finalizzato alla limitazione del traffico peer-to-peer.

L'uso smodato che oggi viene fatto del protocollo P2P, nato per la condivisione di files liberi, spesso si trova in contrasto con la "legge del copyright". La violazione di tale normativa è, in questo contesto, considerata in carico alla coscienza individuale.

Si pone piuttosto l'attenzione sull'influenza negativa che protocolli di questo tipo hanno sulle performance, a causa dell'elevato numero di sessioni generate.

Infine è da ricordare che i programmi di File Sharing sono spesso veicolo di attacchi malevoli e virus. <http://www.sans.org/top20/#a2>

Per tutti questi motivi, spesso è necessario introdurre uno strumento che limiti, specialmente in orari di punta, protocolli di questo tipo.

In questo documento viene presentato uno Shaper che modella il traffico globale peer-to-peer e limita il numero di sessioni contemporaneamente attive per ogni client.

Template disclaimer per le limitazioni di banda del P2P:

Tutela del Diritto d'Autore e Copyright

In conformità a quanto stabilito dalla **Legge 22 aprile 1941, n. 633 - Protezione del diritto d'autore e di altri diritti connessi al suo esercizio** e successiva **Legge 9 gennaio 2008, n. 2 - Disposizioni concernenti la Società italiana degli autori ed editori**, l'*Azienda* dichiara che:

E' demandata alla responsabilità del Cliente l'utilizzo di programmi di condivisione file (Peer to Peer – P2P) nel rispetto delle normative vigenti in materia di **Diritto d'Autore e Copyright**.

L'uso dei programmi Peer to Peer sulla connettività dell'*Azienda* è consentito esclusivamente per la condivisione e distribuzione di materiale non soggetto ai vincoli normativi in materia di **Diritto d'Autore e Copyright**.

Azienda non si ritiene responsabile dell'uso fatto dai propri Clienti di programmi P2P.

Qualità del Servizio

I software di P2P per la condivisione di dati, trasmettono e ricevono costantemente file di grandi dimensioni. Questo genere di attività utilizza molta banda, in entrambe le direzioni di invio e ricezione e può ridurre sensibilmente la velocità dei Clienti che usano altre applicazioni durante le ore di punta.

Inoltre questi protocolli, nella fattispecie eDonkey, generano un numero molto elevato di sessioni con conseguente abbattimento delle performance e sono spesso veicolo di virus e attacchi malevoli.

Per questi motivi, a garanzia della qualità offerta ai propri Clienti (QoS), *Azienda* si riserva la facoltà di adottare, sulle proprie reti, politiche di "Traffic Shaping" (Profilazione del traffico) e "Deep inspection" (Analisi dei pacchetti).

La politica di Qualità del Servizio (QoS) per il corretto utilizzo della banda larga, applicata da *Azienda* ha lo scopo di migliorare la qualità media del servizio a vantaggio di tutti i Clienti.

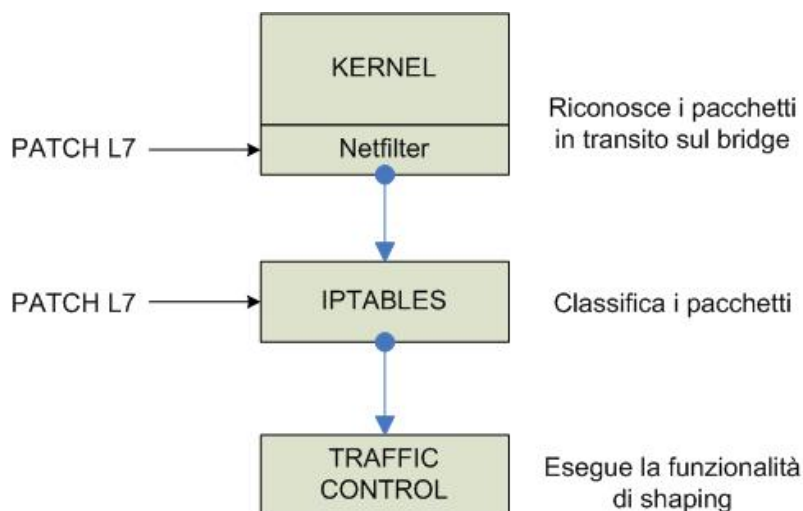
Essa, infatti, non blocca alcun tipo di traffico. Al contrario, privilegia, solo in orari di picco e solo se si rende necessario, il traffico "conversazionale" (HTTP per la navigazione, i protocolli per la gestione della posta elettronica ecc...) rispetto al traffico generato da software di P2P o file sharing.

Fuori dagli orari di punta e quando non è necessario non è attiva alcuna policy di QoS.

IMPLEMENTAZIONE

Lo Shaper proposto per questo scopo è costituito da una macchina su cui sono montate due schede di rete in bridge.

Grazie ad un'apposita patch del kernel i pacchetti che transitano sul bridge vengono riconosciuti e inviati ad Iptables che, a sua volta opportunamente patchato, li classifica e li passa al Traffic Control.



Il Traffic Control ha funzioni di:

- Shaping: controlla il rate di trasmissione;
- Scheduling: programma e priorizza la trasmissione dei pacchetti;
- Policing: crea delle regole sul traffico;
- Dropping: elimina i pacchetti che eccedono la banda predefinita.

Installazione

·1. Creazione del bridge

Si presuppone che le due schede siano ad 1GB, così come il bridge che verrà creato tra eth0 ed eth1.

Quindi, nell' `/etc/rc.local` vanno aggiunte le seguenti linee di comando, per la creazione del bridge:

```
sbin/ifconfig eth0 0.0.0.0 promisc up
/sbin/ifconfig eth1 0.0.0.0 promisc up
echo 1 > /proc/sys/net/ipv4/tcp_tw_recycle
echo "3" > /proc/sys/net/ipv4/tcp_fin_timeout
sysctl -w net.core.rmem_default='8388608'
sysctl -w net.core.wmem_default='8388608'
sysctl -w net.ipv4.tcp_rmem='4096 87380 8388608'
sysctl -w net.ipv4.tcp_wmem='4096 65536 8388608'
sysctl -w net.ipv4.tcp_mem='8388608 8388608 8388608'
```

```
sysctl -w net.ipv4.route.flush=1
/usr/sbin/brctl addbr br0
/usr/sbin/brctl addif br0 eth0
/usr/sbin/brctl addif br0 eth1
ifconfig br0 up promisc
```

·2. Realizzazione delle patch

Innanzitutto si installano i pacchetti propedeutici:

```
apt-get update
apt-get upgrade
apt-get install kernel-package libncurses5-dev fakeroot wget bzip2 g++ g++-4.1 libstdc++6-4.1-dev yaird
```

Dopodiché si acquisiscono i pacchetti relativi al nuovo kernel e ad iptables:

```
cd /usr/src
wget http://ftp.netfilter.org/pub/iptables/iptables-1.4.2.tar.bz2
wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.28.tar.bz2
tar jxvf iptables-1.4.2.tar.bz2
tar jxvf linux-2.6.28.tar.bz2 <http://www.kernel.org/pub/linux/kernel/v2.6/
linux-2.6.28.tar.bz2
```

Si acquisiscono anche:

```
wget http://mesh.dl.sourceforge.net/sourceforge/l7-filter/netfilter-layer7-
v2.21.tar.gz (pacchetto contenente le varie patch)
wget http://kent.dl.sourceforge.net/sourceforge/l7-filter/l7-
protocols-2008-12-18.tar.gz (pacchetto contenente le
definizioni dei protocolli)
tar zxvf netfilter-layer7-v2.21.tar.gz <http://mesh.dl.sourceforge.net/sourceforge/
l7-filter/netfilter-layer7-v2.21.tar.gz
tar zxvf l7-protocols-2008-12-18.tar.gz <http://kent.dl.sourceforge.net/
sourceforge/l7-filter/l7-protocols-2008-12-18.tar.gz
```

Si creano dei link per semplicità di notazione:

```
ln -s linux-2.6.28 linux
ln -s iptables-1.4.2 iptables
```

Si realizza la patch:

```
cd /usr/src/linux
patch -p1 < /usr/src/netfilter-layer7-v2.21/kernel-2.6.25-2.6.28-
layer7-2.21.patch
cd /usr/src/netfilter-layer7-v2.21/iptables-1.4.1.1-for-kernel-2.6.20forward
cp * ../iptables-1.4.2/extensions/
cd /usr/src/linux
make mrproper
```

Attraverso il seguente menù si abilita l'opzione indicata, cosicché ne venga tenuto conto nella creazione del nuovo kernel:

```
make menuconfig
```

```
Networking support [*] Networking options ---> [*] Network packet filtering
framework (Netfilter) ---> [Core Netfilter Configuration ---> <M> "layer7"
match support
```

A questo punto, si crea il pacchetto contenente il nuovo kernel:

```
make-kpkg clean
make-kpkg --revision hbdebl7 --append-to-version .20081218 --initrd binary-
arch
```

E lo si installa:

```
cd /usr/src/
dpkg -i linux-image-2.6.28.20081218_debl7_i386.deb
```

Si verifica che sia il nuovo kernel ad essere caricato al boot:

```
vi /boot/grub/menu.lst
il valore di default deve corrispondere al kernel appena compilato 2.6.28
reboot
```

A questo punto è possibile installare iptables, scaricato in precedenza:

```
cd /usr/src/iptables
./configure --with-ksource=/usr/src/linux
make
make install
```

Ed infine installare le definizioni dei protocolli, precedentemente acquisite:

```
tar zxvf l7-protocols-2008-12-18.tar.gz
cd l7-protocols-2008-12-18
make install
```

·3. Definizione delle regole

Oltre ai bridge fisici, vanno definiti i flussi di pacchetti da analizzare ed i filtri con cui effettuare tale analisi. Tutto questo è dichiarato in un file posto in */root* e denominato *regole*.

In questo particolare esempio si suppone di dover limitare il traffico generato dai protocolli: bittorrent, irc, gnutella, edonkey e fasttrack.

Il traffico non viene droppato, al raggiungimento della soglia prefissata, ma viene declassato e gli vengono assegnati parametri meno favorevoli, così da scoraggiarlo.

Si generano le catene e si definisce il traffico che vi scorre:

```
/usr/local/sbin/iptables -t mangle -N ms-all
/usr/local/sbin/iptables -t mangle -N ms-all-chains
/usr/local/sbin/iptables -t mangle -N ms-prerouting
```

```
/usr/local/sbin/iptables -t mangle -A PREROUTING -j ms-prerouting
/usr/local/sbin/iptables -t mangle -A ms-prerouting -j CONNMARK --restore-
mark
```

Si definisce il brigde di traffico in ingresso (eth1 -> eth0):

```
/sbin/tc qdisc add dev eth0 handle 1: root htb default 1
/usr/local/sbin/iptables -t mangle -A ms-prerouting -m physdev --physdev-in eth1
-j ms-all
/usr/local/sbin/iptables -t mangle -A POSTROUTING -m physdev --physdev-out
eth0 -j ms-all-chains
/sbin/tc class add dev eth0 parent 1: classid 1:1 htb rate 1048576Kbit
/sbin/tc filter add dev eth0 parent 1:0 protocol all u32 match u32 0 0 classid 1:1
```

Viene definita la catena contenente il traffico precedente a cui vengono assegnati determinati parametri:

```
##### Incoming Rules
##### chain Speedy
/sbin/tc class add dev eth0 parent 1:1 classid 1:11 htb rate 102400Kbit ceil
102400Kbit quantum 1532
/usr/local/sbin/iptables -t mangle -N ms-chain-eth0-1:11
/usr/local/sbin/iptables -t mangle -A ms-all-chains -m connmark --mark
0x6ae4889e -j ms-chain-eth0-1:11
/usr/local/sbin/iptables -t mangle -A ms-all -m physdev --physdev-in eth1 -j
MARK --set-mark 0x6ae4889e
/usr/local/sbin/iptables -t mangle -A ms-all -m physdev --physdev-in eth1 -j
RETURN
```

Si dichiarano i filtri secondo cui selezionare il traffico da declassare nella classe definita di seguito, logicamente figlia della precedente:

```
##### generating pipes for Speedy
##### pipe P2P
```

Definizione della classe con priorità inferiore:

```
/sbin/tc class add dev eth0 parent 1:11 classid 1:12 htb rate 6554Kbit ceil
6554Kbit burst 0.01Kbit quantum 1532
/sbin/tc qdisc add dev eth0 handle 12: parent 1:12 sfq
```

Definizione dei filtri che determinano le pipes:

```
/usr/local/sbin/iptables -t mangle -A ms-chain-eth0-1:11 -m layer7 --l7proto
bittorrent -j CLASSIFY --set-class 1:12
/usr/local/sbin/iptables -t mangle -A ms-chain-eth0-1:11 -m layer7 --l7proto
bittorrent -j RETURN
```

```
/usr/local/sbin/iptables -t mangle -A ms-chain-eth0-1:11 -m layer7 --l7proto irc
-j CLASSIFY --set-class 1:12
```

```
/usr/local/sbin/iptables -t mangle -A ms-chain-eth0-1:11 -m layer7 --l7proto irc
-j RETURN
```

```
/usr/local/sbin/iptables -t mangle -A ms-chain-eth0-1:11 -m layer7 --l7proto
gnutella -j CLASSIFY --set-class 1:12
```

```

/usr/local/sbin/iptables -t mangle -A ms-chain-eth0-1:11 -m layer7 --l7proto
gnutella -j RETURN
/usr/local/sbin/iptables -t mangle -A ms-chain-eth0-1:11 -m layer7 --l7proto
edonkey -j CLASSIFY --set-class 1:12
/usr/local/sbin/iptables -t mangle -A ms-chain-eth0-1:11 -m layer7 --l7proto
edonkey -j RETURN
/usr/local/sbin/iptables -t mangle -A ms-chain-eth0-1:11 -m layer7 --l7proto
fasttrack -j CLASSIFY --set-class 1:12
/usr/local/sbin/iptables -t mangle -A ms-chain-eth0-1:11 -m layer7 --l7proto
fasttrack -j RETURN

```

Viene conclusa la catena del traffico in ingresso:

```

/sbin/tc class add dev eth0 parent 1:11 classid 1:199 htb rate 102400Kbit ceil
102400Kbit quantum 1532
/sbin/tc qdisc add dev eth0 handle 199: parent 1:199 sfq
/usr/local/sbin/iptables -t mangle -A ms-chain-eth0-1:11 -j CLASSIFY --set-class
1:199
/usr/local/sbin/iptables -t mangle -A ms-chain-eth0-1:11 -j RETURN

```

Si osserva che la dimensione complessiva delle pipes contenute in una catena non può superare la dimensione complessiva della stessa.

In modo del tutto analogo si definiscono chain, filtri e pipes relativi al traffico in uscita:

Si definisce il brigde di traffico in uscita (eth0 -> eth1):

```

/sbin/tc qdisc add dev eth1 handle 1: root htb default 1
/usr/local/sbin/iptables -t mangle -A ms-prerouting -m physdev --physdev-in eth0
-j ms-all
/usr/local/sbin/iptables -t mangle -A POSTROUTING -m physdev --physdev-out
eth1 -j ms-all-chains
/sbin/tc class add dev eth1 parent 1: classid 1:1 htb rate 1048576Kbit
/sbin/tc filter add dev eth1 parent 1:0 protocol all u32 match u32 0 0 classid 1:1

```

Viene definita la catena contenente il traffico precedente a cui vengono assegnati determinati parametri:

```

##### Outgoing Rules
##### chain Speedy
/sbin/tc class add dev eth1 parent 1:1 classid 1:11 htb rate 102400Kbit ceil
102400Kbit quantum 1532
/usr/local/sbin/iptables -t mangle -N ms-chain-eth1-1:11
/usr/local/sbin/iptables -t mangle -A ms-all-chains -m connmark --mark
0xd258eff9 -j ms-chain-eth1-1:11
/usr/local/sbin/iptables -t mangle -A ms-all -m physdev --physdev-in eth0 -j
MARK --set-mark 0xd258eff9
/usr/local/sbin/iptables -t mangle -A ms-all -m physdev --physdev-in eth0 -j
RETURN

```

Si dichiarano i filtri secondo cui selezionare il traffico da declassare nella classe definita di seguito, logicamente figlia della precedente:

```

##### generating pipes for Speedy

```

```
##### pipe P2P
```

Definizione della classe con priorità inferiore:

```
/sbin/tc class add dev eth1 parent 1:11 classid 1:12 htb rate 6554Kbit ceil  
6554Kbit burst 0.01Kbit quantum 1532  
/sbin/tc qdisc add dev eth1 handle 12: parent 1:12 sfq
```

Definizione dei filtri che determinano le pipes:

```
/usr/local/sbin/iptables -t mangle -A ms-chain-eth1-1:11 -m layer7 --l7proto  
bittorrent -j CLASSIFY --set-class 1:12  
/usr/local/sbin/iptables -t mangle -A ms-chain-eth1-1:11 -m layer7 --l7proto  
bittorrent -j RETURN  
/usr/local/sbin/iptables -t mangle -A ms-chain-eth1-1:11 -m layer7 --l7proto irc  
-j CLASSIFY --set-class 1:12  
/usr/local/sbin/iptables -t mangle -A ms-chain-eth1-1:11 -m layer7 --l7proto irc  
-j RETURN  
/usr/local/sbin/iptables -t mangle -A ms-chain-eth1-1:11 -m layer7 --l7proto  
gnutella -j CLASSIFY --set-class 1:12  
/usr/local/sbin/iptables -t mangle -A ms-chain-eth1-1:11 -m layer7 --l7proto  
gnutella -j RETURN  
/usr/local/sbin/iptables -t mangle -A ms-chain-eth1-1:11 -m layer7 --l7proto  
edonkey -j CLASSIFY --set-class 1:12  
/usr/local/sbin/iptables -t mangle -A ms-chain-eth1-1:11 -m layer7 --l7proto  
edonkey -j RETURN  
/usr/local/sbin/iptables -t mangle -A ms-chain-eth1-1:11 -m layer7 --l7proto  
fasttrack -j CLASSIFY --set-class 1:12  
/usr/local/sbin/iptables -t mangle -A ms-chain-eth1-1:11 -m layer7 --l7proto  
fasttrack -j RETURN
```

Viene conclusa la catena del traffico in uscita:

```
/sbin/tc class add dev eth1 parent 1:11 classid 1:199 htb rate 102400Kbit ceil  
102400Kbit quantum 1532  
/sbin/tc qdisc add dev eth1 handle 199: parent 1:199 sfq  
/usr/local/sbin/iptables -t mangle -A ms-chain-eth1-1:11 -j CLASSIFY --set-class  
1:199  
/usr/local/sbin/iptables -t mangle -A ms-chain-eth1-1:11 -j RETURN  
/usr/local/sbin/iptables -t mangle -A ms-prerouting -j CONNMARK --save-mark  
/usr/local/sbin/iptables -A FORWARD -m layer7 --l7proto edonkey -m  
connlimit --connlimit-above 100 -j REJECT  
/usr/local/sbin/iptables -A FORWARD -j ACCEPT
```

Si osserva che la dimensione complessiva delle pipes contenute in una catena non può superare la dimensione complessiva della stessa.

In conclusione al documento può essere utile notare che per eliminare lo shaping (e quindi cancellare chains, pipes e filtri) è possibile utilizzare uno script analogo a questo:

```
/usr/local/sbin/iptables -t mangle -F  
for table in nat filter mangle
```

```
do
  /usr/local/sbin/iptables -t $table -F #svuota le catene
  /usr/local/sbin/iptables -t $table -X #cancella le catene
done
/sbin/tc qdisc del dev eth1 root
/sbin/tc qdisc del dev eth0 root
```