UNIVERSITÀ DEGLI STUDI DI CAMERINO

FACOLTÀ DI SCIENZE E TECNOLOGIE

Corso di Laurea in Informatica

Dipartimento di Matematica e Informatica

REALIZZAZIONE DI UN SISTEMA DI TRAFFIC-SHAPING L7-FILTER

Tesi di Laurea sperimentale
In Reti di Elaboratori

Laureando

Stefano Gironella

Relatore

Dott. Fausto Marcantoni

ANNO ACCADEMICO 2007 / 2008

Ringraziamenti

Desidero innanzitutto ringraziare il professore Fausto Marcantoni, relatore di questa tesi, per la grande disponibilità e cortesia dimostratemi, e per tutto l'aiuto fornito durante la stesura.

Un sentito ringraziamento ai miei genitori, che, con il loro incrollabile sostegno morale ed economico, mi hanno permesso di raggiungere questo traguardo.

Un ultimo ringraziamento a Carla e tutti i miei amici per essermi stati vicini sia nei momenti difficili, sia nei momenti felici.

INDICE

1. INTRODUZIONE

L'evoluzione dei servizi offerti da internet, accompagnata dal sempre maggior numero di utenti che ne usufruiscono, ha portato alla crescita esponenziale dell'uso delle risorse di rete.

Servizi multimediali *on-demand* e *peer to peer* tendono a saturare rapidamente la banda disponibile con conseguenze negative sul resto delle applicazioni che necessitano di internet; questo porta alla necessità di gestire il traffico della rete al fine di ottimizzare l'uso delle risorse disponibili e impedendo, al tempo stesso, che gli utenti ne abusino violando le normative legali vigenti.

L'utilizzo di firewall (software o tramite hardware dedicato) è divenuto necessario, anche in reti di medio-piccole dimensioni, per poter identificare e gestire il traffico dati.

1.1 Scopo del progetto

L'obiettivo di questo progetto è implementare un firewall che operi a livello applicativo, al fine di dare una priorità al traffico delle diverse applicazioni che comunicano tramite Internet, limitando (o bloccando) la banda disponibile per le applicazioni non fondamentali e, al tempo stesso, rendere la configurazione dello stesso semplice anche per utenti con basse conoscenze tecniche.

1.2 Perché un firewall di livello 7

Un firewall di questo tipo, a differenza di uno di livello 3, non effettua controlli sui pacchetti di dati analizzando gli indirizzi Ip, le porte utilizzate o il tipo di protocollo utilizzato; bensì legge il contenuto del payload del pacchetto confrontandolo con le espressioni regolari definite nei pattern di cui dispone allo scopo di identificare l'applicazione che lo ha generato.

Attualmente sono disponibili due ottimi classificatori, entrambi integrabili come estensione di Netfilter, il firewall standard di Linux:

- IPP2P (http://www.ipp2p.org/)
- L7-Filter (http://l7-filter.sourceforge.net/)

Il primo è finalizzato alla gestione dei più diffusi applicativi peer to peer, il secondo mette a disposizione un set di pattern più numeroso e comprendete applicazioni di vario tipo.

È stato scelto L7-Filter per via del maggior numero di programmi riconosciuti, inoltre lo stesso viene utilizzato in numerose distribuzioni Linux finalizzate alla gestione della rete (Zeroshell, Untangle, eBox, ecc.), è open source, si integra facilmente in qualsiasi distribuzione Linux ed è accompagnato da una comunità di sviluppo molto numerosa.

2 QOS

Il termine QOS (Quality of Service) sta ad indicare la capacità di una architettura di rete di gestire in maniera differenziata differenti tipologie di traffico.

Gli strumenti, tramite il quale è possibile implementare politiche di QOS, ci consentono di poter "privilegiare" applicazioni che necessitano di basse latenze a discapito di quelle per cui un leggero ritardo nell'invio/ricezione dei dati può essere ininfluente, stabilire una soglia massima per la banda messa a disposizione o una minima, sotto la quale non si vuole scendere.

In questo progetto si è voluto rendere possibile la gestione del traffico di rete

- Bloccandolo
- Permettendolo
- Assegnando una priorità di instradamento alta o bassa
- Scegliendo una tra le cinque possibilità di assegnamento della banda (75%, 50%, 25%, 10% e 5% del massimo disponibile)

2.1 L7-Filter

L7-Filter è un classificatore di pacchetti per Linux; è in grado di eseguire un confronto sul contenuto del payload del pacchetti a livello applicativo con dei pattern predefiniti, allo scopo di individuare l'applicazione che li ha generati.

Il motivo principale per utilizzare L7-Filter è facilitare la gestione del traffico generato da applicazioni che utilizzano porte non standard o comunque non prevedibili come molti programmi P2P.

La classificazione avviene analizzando il payload dei primi 10 pacchetti (o i primi 2 kB di dati) relativi ad ogni nuova connessione; il numero di pacchetti da analizzare può essere comunque aumentato o diminuito agendo sul file layer7_numpackets nella directory /proc/net/.

L'analisi è mirata ai soli pacchetti iniziali (quindi non a tutti i pacchetti relativi alla connessione) in quanto si sfruttano le capacità di connection tracking di linux; in pratica non viene gestito il pacchetto in se, ma i pacchetti relativi ad una certa connessione, diminuendo quindi il numero di controlli che altrimenti dovrebbero essere effettuati dal firewall.

Essendo L7-Filter un'estensione di Netfilter, questo classificatore può essere utilizzato per qualsiasi operazione effettuabile con il firewall di Linux implementando le regole tramite Iptables.

È quindi possibile, per ogni applicazione riconosciuta:

- Bloccare i pacchetti relativi a tale applicazione.
- Assegnare una banda massima / minima garantita.
- Impostare una bassa / alta priorità di instradamento.
- Effettuare statistiche sul numero dei pacchetti inviati/ricevuti

Il problema principale di questo tipo di classificazione è legato all'uso dei pattern per riconoscere i protocolli; se da una parte questo semplifica di molto la definizione delle regole per la gestione del traffico nella rete, dall'altra si crea la possibilità di falsi positivi nel controllo dei pacchetti.

Ogni pattern è relativo ad un protocollo riconosciuto e contiene una stringa da confrontare con il payload del pacchetto analizzato, se tale stringa è presente tutti i pacchetti relativi a quella connessione vengono gestiti in base alle regole che sono state definite.

Poniamo di aver definito una regola per gestire il traffico dell'applicazione A; l'applicazione B tenta di effettuare una connessione e i suoi primi pacchetti vengono analizzati, l'esito dell'analisi è un match con il pattern relativo all'applicazione A (falso positivo), conseguentemente tutti i pacchetti generati dall'applicazione B vengono erroneamente gestiti come se appartenessero ad A.

Se la regola definita per gestire il traffico generato da A prevede di bloccarlo, l'applicazione B non riuscirebbe ad inviare / ricevere pacchetti nonostante il suo uso fosse legittimo.

È consigliabile quindi usare questo classificatore per gestire il traffico applicando regole di traffic-shaping, che comporterebbero magari una riduzione della banda utilizzabile o una bassa priorità di instradamento, consentendo comunque (nel caso di falso positivo) all'applicazione di accedere alla rete piuttosto che bloccarla.

2.2 I pattern

I pattern definiscono il nome del protocollo a cui si riferiscono e l'espressione regolare da ricercare nei pacchetti dati.

Nella figura 2.1 si può vedere il contenuto del pattern relativo al protocollo edonkey.

```
# eDonkey2000 - P2P filesharing - http://edonkey2000.com and others
# Pattern attributes: good veryfast fast overmatch
# Protocol groups: p2p
# Wiki: http://www.protocolinfo.org/wiki/EDonkey
# Copyright (C) 2008 Matthew Strait, Ethan Sommer; See ../LICENSE
#
# Tested recently (April/May 2006) with eMule 0.47a and eDonkey2000 1.4
# and a long time ago with something else.
#
# In addition to matching what you might expect, this matches much of
# what eMule does when you tell it to only connect to the KAD network.
# I don't quite know what to make of this.

# Thanks to Matt Skidmore <fox AT woozle.org>

edonkey

# http://gd.tuwien.ac.at/opsys/linux/sf/p/pdonkey/eDonkey-protocol-0.6
#
# In addition to \xe3, \xc5 and \xd4, I see a lot of \xe5.
# As of April 2006, I also see some \xe4.
#
# God this is a mess.  What an irritating protocol.
# This will match about 2% of streams with random data in them!
# (But fortunately much fewer than 2% of streams that are other protocols.
# You can test this with the data in ../testing/)

^[\xc5\xd4\xe3-\xe5].?.?.?([\x01\x02\x05\x14\x15\x16\x18\x19\x1a\x1b\x1c\x20\x21\x32\x33\x34
\x35\x36\x38\x40\x41\x42\x43\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x
56\x57\x58[\x60\x81\x82\x90\x91\x93\x96\x97\x98\x99\x9a\x9b\x9c\x9e\xa0\xa1\xa2\xa3\xa4]|\x59.
..............?[ -~]|\x96....$)

# matches everything and too much
# ^(\xe3|\xc5|\xd4)

# ipp2p essentially uses "\xe3....\x47", which doesn't seem at all right to me.

# bandwidtharbitrator uses
# e0.*@.*6[a-z].*p$|e0.*@.*[a-z]6[a-z].*p0$|e.*@.*[0-9]6.*p$|emule|edonkey
# no comments to explain what all the mush is, of course...
```

(Figura 2.1 – Pattern edonkey)

La prima parte non commentata è quella contenente il nome del protocollo (edonkey), la seconda contiene l'espressione regolare.

I commenti riportano diverse informazioni utili alla comprensione del protocollo a cui fa riferimento il pattern:

- Nome e descrizione sommaria del protocollo
- Info sul pattern
    - Qualità:
    - Great: Funziona correttamente (sono stati effettuati molti test).
    - Good: Funziona correttamente (sono stati effettuati sufficienti test).
    - Ok: Dovrebbe funzionare.
    - Marginal: Può non funzionare.
    - Poor: Probabilmente non funziona.
    - Velocità (Sistema di riferimento con processore Pentium 3 450MHz):
    - Very fast: 0.8–2 secondi.
    - Fast: 2–8 secondi.
    - Not so fast: 8–100 secondi.
    - Slow: più di 100 secondi.
    - Altro:
    - Overmatching: può comportare il verificarsi di falsi positivi.

  - Undermatching: può non essere in grado di riconoscere tutti i pacchetti ma solo alcuni di essi.
  - Superset: Effettua il match con un set di pacchetti identificati singolarmente da pattern specifici.
  - Subset: Effettua il match con pacchetti identificabili da pattern maggiormente generici.

- Tipologia del protocollo – chat, p2p, streaming_video, ecc.
- Link ad una o più pagine web descriventi il protocollo

I pattern disponibili in data 23/11/ 2008:

100bao
aim
aimwebcontent
applejuice
ares
armagetron
battlefield1942
battlefield2142
battlefield2
bgp
biff
bittorrent
chikka
cimd
ciscovpn
citrix
counterstrike-source
cvs
dayofdefeat-source
dhcp
directconnect
dns
doom3
edonkey
fasttrack
finger
freenet
ftp
gkrellm
gnucleuslan
gnutella
goboogy
gopher
guildwars
h323
halflife2-deathmatch
hddtemp
hotline
http
http-rtsp
ident
imap
imesh
ipp
irc
jabber
kugoo
live365
liveforspeed
lpd
mohaa
msn-filetransfer
msnmessenger
mute
napster
nbns
ncp
netbios
nntp
ntp
openft
pcanywhere
poco
pop3
pplive
qq
quake1
quake-halflife
radmin
rdp
replaytv-ivs
rlogin
rtp
rtsp
shoutcast
sip
skypeout
skypetoskype
smb
smtp
snmp
socks
soribada
soulseek
ssdp
ssh
ssl
stun
subspace
subversion
teamfortress2
teamspeak
telnet
tesla
tftp
thecircle
tor
tsp
unknown
unset
uucp
validcertssl
ventrilo
vnc
whois
worldofwarcraft
x11
xboxlive
xunlei
yahoo
zmaap

2.3 Netfilter ed Iptables

Nei kernel 2.4.x e 2.6.x di Linux le attività di firewall sono implementate per mezzo del framework Netfilter.

Netfilter è il tool che ci consente di gestire, analizzare e manipolare i pacchetti in arrivo, in transito ed in uscita dalle interfacce di rete.

Iptables è il front-end tramite il quale definiamo il comportamento di Netfilter.

Inizialmente nessuna regole è attiva, quindi tutto il traffico dati è consentito; una volta che le regole sono state inserite, il kernel controlla tutti i pacchetti passanti per le interfacce di rete e, se il pacchetto corrisponde ad uno di quelli per cui è stata definita una regola, esegue l'azione stabilita.

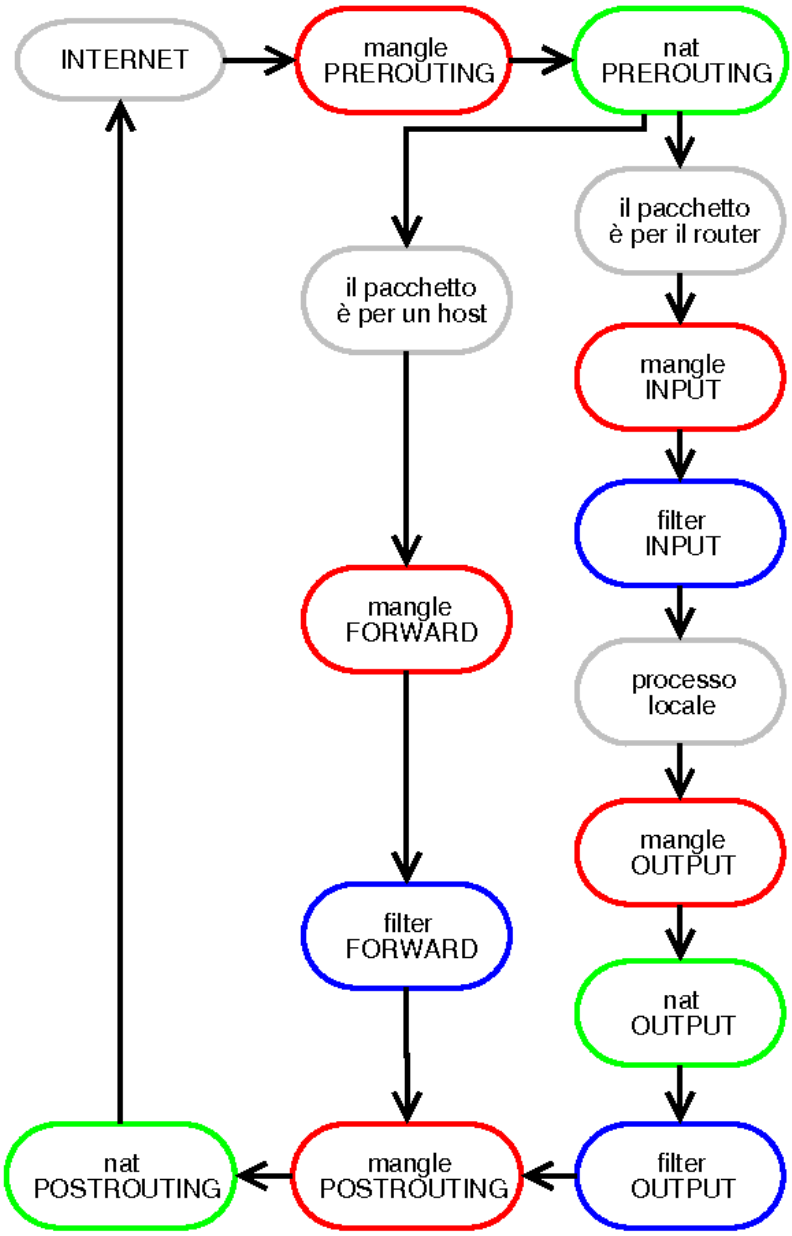
Di default Netfilter dispone di tre tabelle, ognuna delle quali contiene un set di regole (chain) predefinite (figura 2.2).

Tabelle:

- Nat – per definire regole che comportano la modifica nei pacchetti degli indirizzi ip, porte, ecc.
- Filter – per definire come gestire i pacchetti (accettarli, scartarli, rifiutarli, ecc).
- Mangle – per alterare il contenuto dei pacchetti.

Chains:

- INPUT – regole applicabili ai pacchetti destinati alla macchina stessa.
- OUTPUT – regole applicabili ai pacchetti generati dalla macchina stessa.
- PREROUTING – regole applicabili ai pacchetti in entrata prima di consultare la tabella di routing.
- POSTROUTING – regole applicabili ai pacchetti in uscita dopo aver consultato la tabella di routing.
- FORWARD – regole applicabili ai pacchetti da inoltrare ad un altro indirizzo IP.



(Figura 2.2 – Schema Iptables)

All'interno del progetto, Netfilter è stato utilizzato per implementare:

- NAT.
- Blocco di un protocollo.
- Mark di un protocollo.

Bloccando un protocollo definiamo una regola per Netfilter che scarta i pacchetti relativi alla connessione identificata tramite i pattern l7-filter impedendogli, di fatto, di comunicare.

Es:

```
iptables -A FORWARD -m layer7 --l7proto nome_protocollo -
j DROP
```

Mark prevede l'assegnazione di un identificativo numerico ai pacchetti relativi ad una connessione; successivamente, tramite Traffic control, i pacchetti vengono identificati in base al numero che gli è stato assegnato e gestiti dalle relative regole di accodamento

Es:

```
iptables -t mangle -A PREROUTING -i eth0 -m layer7 --
l7proto nome_protocollo -j MARK --set-mark 25
```

2.4 Traffic control (TC)

Traffic control, tool appartenente alla suite Iproute2, ci permette di definire la banda disponibile e la priorità di instradamento per un certo tipo di traffico dati in base alle nostre necessità.

Di default la disciplina di accodamento dei pacchetti è di tipo FIFO (first in first out) e non vengono applicate limitazioni di banda; nell'implementazione di questo firewall è stata invece utilizzata la disciplina HTB (Hierarchical Token Bucket).

L'idea di base è definire, per ogni interfaccia di rete, una classe root dalla quale dipendono delle sotto classi, ognuna caratterizzata da:

- Un identificativo univoco.
- L'identificativo della classe principale.
- Banda massima / minima garantita.
- Priorità di instradamento.

Tramite le sottoclassi definiremo le diverse discipline di accodamento che dovranno essere applicate per i pacchetti relativi alle applicazioni classificate tramite L7-Filter.

Un esempio:

Definiamo la disciplina di accodamento (qdisc) root per l'interfaccia di rete eth0 e specifichiamo che tutti i pacchetti per i quali non sono previste regole specifiche vengano gestiti dalla sottoclasse 1:10.

```
tc qdisc add dev eth0 root handle 1: htb default 10
```

Definiamo la classe root relativa alla qdisc appena creata ed indichiamo la banda minima garantita e quella massima.

```
tc class add dev eth0 parent 1: classid 1:1 htb rate
720kbit ceil 800kbit prio 1
```

Creiamo due sottoclassi.

```
tc class add dev eth0 parent 1:1 classid 1:10 htb rate
720kbit ceil 800kbit prio 1
tc  class add dev eth0 parent 1:1 classid 1:11 htb rate
540kbit ceil 600kbit prio 1
```

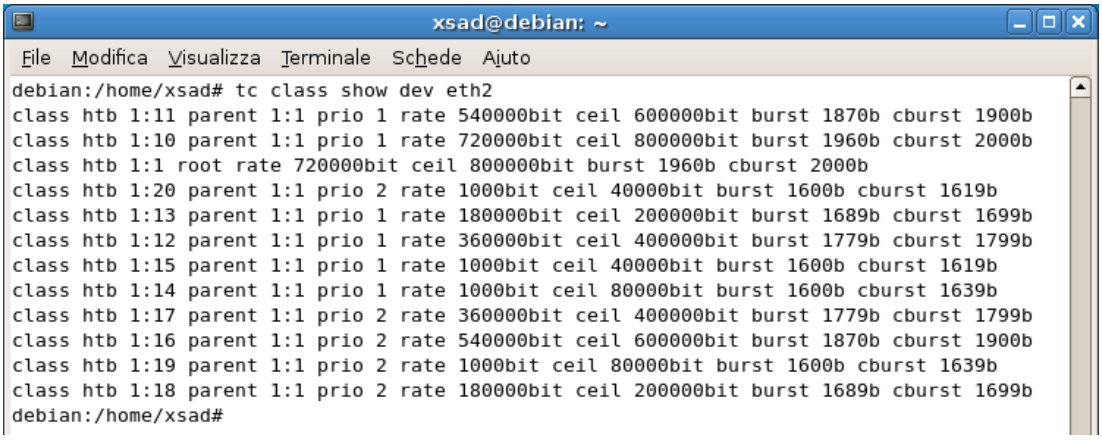
Definiamo quali pacchetti dovranno essere accodati ed in quale sottoclasse

```
tc filter add dev eth0 protocol ip parent 1:0 prio 1
handle 100 fw flowid 1:10

tc filter add dev eth0 protocol ip parent 1:0 prio 5
handle 75 fw flowid 1:11
```

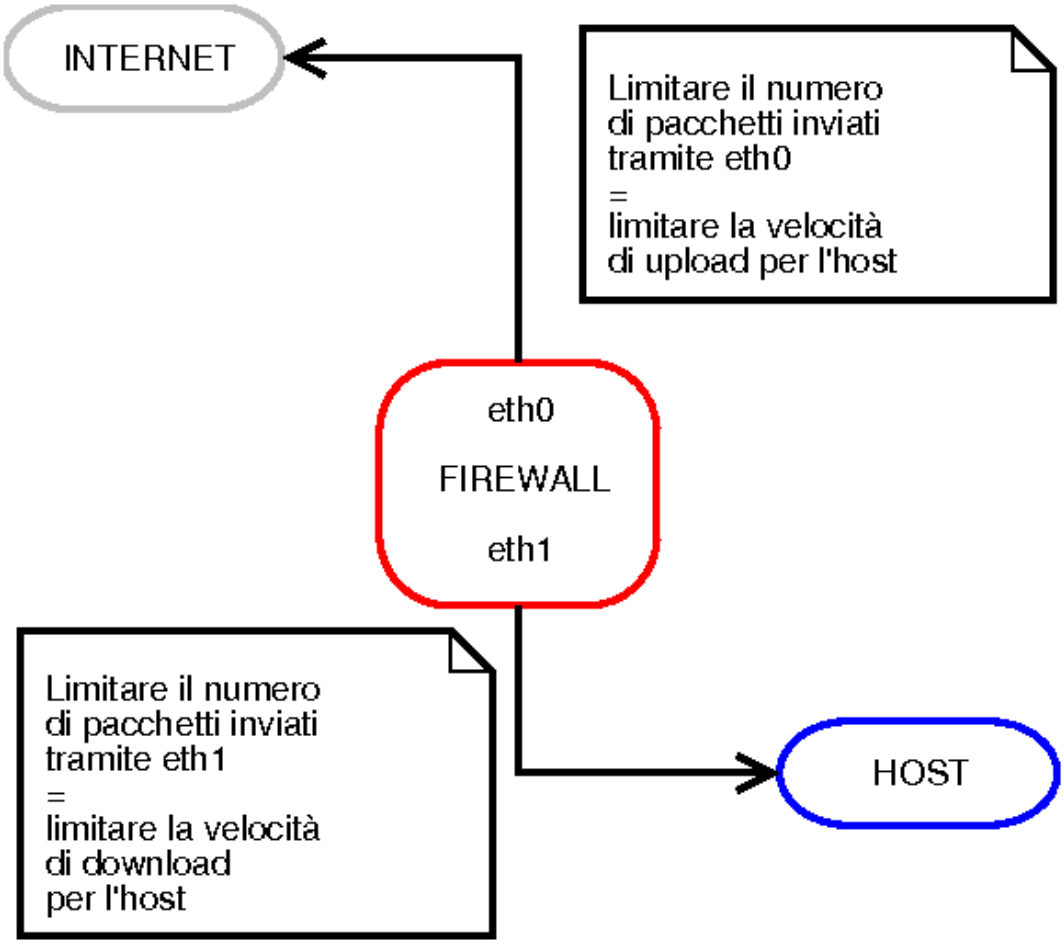
I valori "*handle …*" sono collegati al mark dei pacchetti effettuato da netfilter; in questo caso abbiamo impostato che i pacchetti marcati con "100" vengano gestiti dalla classe 1:10 e quelli marcati con "75" dalla classe 1:11.

Tramite il comando `tc class show dev nome_interfaccia` è possibile verificare il corretto inserimento delle regole:



(Figura 2.3 – Visualizzazione configurazione TC)

Definire una banda massima / minima per una connessione significa "rallentare" di quanto basta l'instradamento dei pacchetti (figura 2.4).



(Figura 2.4 – Traffic-shaping, schema logico)

3. INSTALLAZIONE E CONFIGURAZIONE DEL SISTEMA

L'implementazione del sistema ha previsto due fasi:

- Abilitazione del supporto ad L7-Filter nel kernel ed in Iptables.
- Implementazione dell'interfaccia web di gestione

Attualmente sono disponibili due versioni del classificatore L7-Filter:

- Kernel version.
- Userspace version.

La kernel version è quella più stabile e ben testata ma, di contro, richiede la compilazione di kernel e Iptables applicando le patch fornite nel pacchetto "netfilter-layer7" disponibile sul sito dello sviluppatore.

La versione userspace consente un'installazione più semplice ma non è ancora definita sufficientemente stabile.

3.1 Prerequisiti

1. Una distribuzione di linux.
2. Il pacchetto netfilter-layer7 (reperibile da http://l7-filter.sourceforge.net/).
3. Il pacchetto contenente le definizioni dei protocolli riconosciuti (reperibile da http://l7-filter.sourceforge.net/).
4. I sorgenti di un kernel compatibile (reperibile da http://kernel.org/).

   Attualmente sono supportate le versioni 2.6.25.

   - 2.6.22 -> 2.6.24.
   - 2.6.20 -> 2.6.21.
   - 2.6.18 -> 2.6.19.
   - 2.6.17.
   - 2.6.13-> 2.6.16.
   - 2.6.11-> 2.6.12.
   - 2.6.9 - 2.6.10.
   - 2.6.0 -> 2.6.8.1.
   - 2.4.

5. I sorgenti di una versione compatibile di Iptables (reperibile da http://netfilter.org/).

   Attualmente sono supportate le versioni:

   - 1.4.1.1 per kernel 2.6.20 e successivi.
   - 1.4 per kernel 2.6.20 e successivi.
   - 1.3 per kernel 2.6.20 e successivi.
   - 1.3 per kernel precedenti al 2.6.20.

Nel mio caso i test sono stati effettuati utilizzando inizialmente Debian v.4 e successivamente CentOS v.5.2 per accertare il corretto funzionamento su due delle distribuzioni Linux più diffuse.

Essendo il firewall eseguito su Linux, ho preferito utilizzare esclusivamente tool open source o comunque gratuiti; l'interfaccia web di gestione è composta da pagine Jsp eseguite dal web application server Apache Tomcat, lo sviluppo di tali pagine è stato eseguito utilizzando l'IDE Netbeans 6.5 e la gestione delle connessioni ssh è stata implementata utilizzando la libreria Trilead ssh.

3.2 Abilitare il supporto ad L7-Filter

La seguente procedura utilizza:

- Distribuzione Debian 40r5
- Netfilter-layer7-v2.20
- L7-protocols-2008-12-18
- Kernel 2.6.25
- Iptables 1.4

Per quanto riguarda le altre distribuzioni, l'abilitazione del supporto ad L7-Filter avviene in modo analogo, adattando esclusivamente i comandi relativi alla compilazione dei pacchetti (Kernel ed Iptables).

1. Installare il sistema operativo.
2. Terminata l'installazione, autenticarsi come utente root e, all'interno di una directory (nel mio caso /usr/src), scaricare:

2.1 Sorgenti del kernel

(http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.25.tar.bz2).

2.2 Sorgenti Iptables

(http://netfilter.org/projects/iptables/files/iptables-1.4.0.tar.bz2).

2.3 Patch

(http://freefr.dl.sourceforge.net/sourceforge/l7-filter/netfilter-layer7-v2.20.tar.gz).

2.4 Definizioni protocolli

(http://freefr.dl.sourceforge.net/sourceforge/l7-filter/l7-protocols-2008-12-18.tar.gz).

3. Scompattare i sorgenti del kernel all'interno della directory e creare un link simbolico che punti alla directory appena creata

```
tar xvjf linux-source-2.6.25.tar.bz2

ln –s linux-source-2.6.25 linux
```

4. Scompattare il pacchetto contenente le patch

```
tar xvzf netfilter-layer7-v2.20.tar.gz
```

5. Spostarsi all'interno della directory /usr/src/linux (il link simbolico creato precedentemente) ed applicare la patch al kernel

```
cd /usr/src/linux

patch -p1 < /usr/src/netfilter-layer7-v2.20/kernel-2-6-25-layer7-2.20.patch
```

6. Eseguire una delle utility per configurare il kernel (es. menuconfig) ed abilitare il supporto ad L7 Filter (eventualmente abilitare tutte le voci relative a Netfilter)

```
make menuconfig
```

7. Salvare la configurazione ed eseguire la compilazione del kernel

```
make-kpkg --append-to-version=-L7 -initrd kernel_image
```

8. Al termine del processo di compilazione, all'interno della directory /usr/src/ sarà presente il nuovo kernel.
9. Installarlo e riavviare il computer
10. Tornare nella directory /usr/src, scompattare i sorgenti di Iptables all'interno della directory e creare un link simbolico che punti alla directory appena creata

```
tar xvjf iptables-1.4.0.tar.bz2

ln -s iptables-1.4.0 iptables
```

11. Spostarsi all'interno della directory /usr/src/iptables (il link simbolico creato precedentemente) ed applicare la patch ad Iptables

```
patch -p1 < /usr/src/netfilter-layer7-v2.20/iptables-1.4-for-kernel-2.6.20forward-layer7-2.20.patch
```

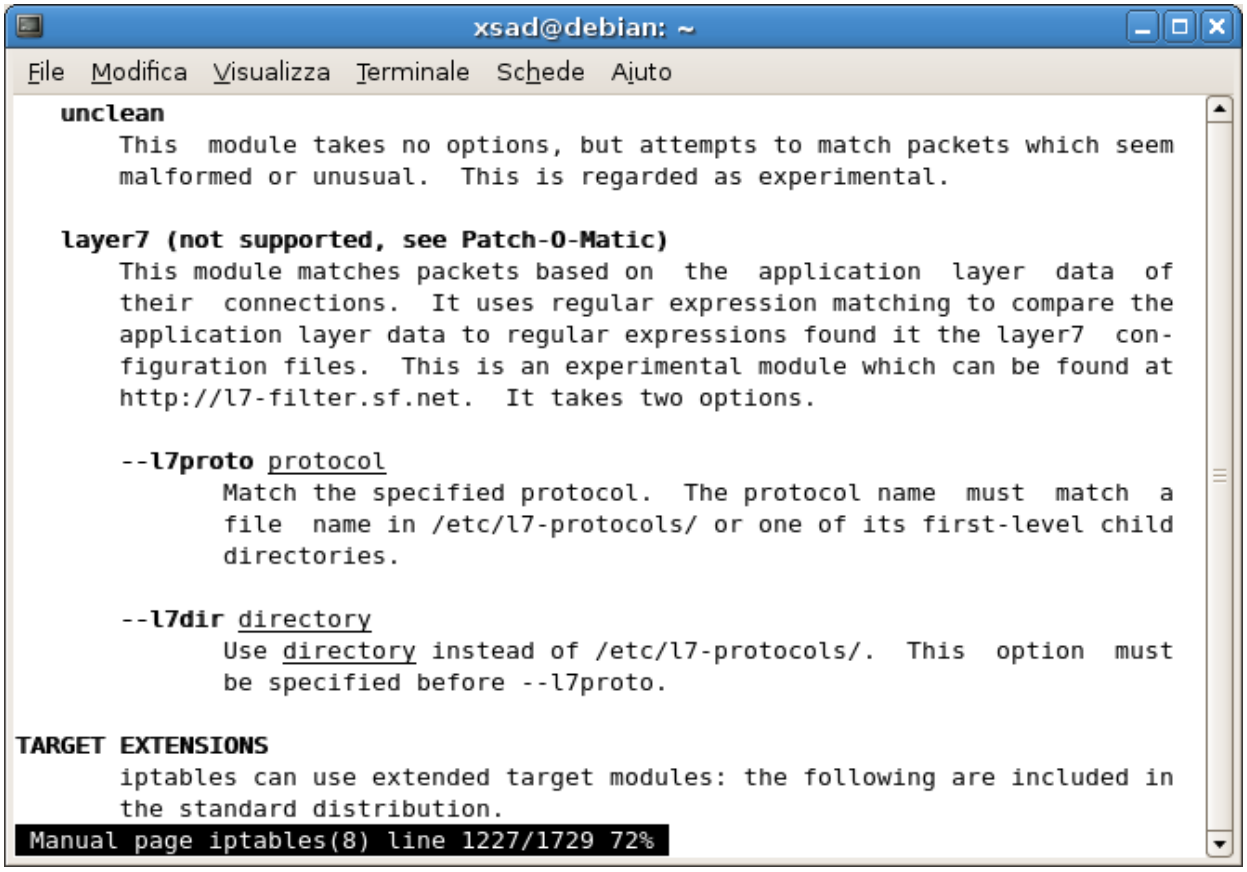
12. Rendere eseguibile l'estensione L7 e compilare Iptables

```
chmod +x extensions/.layer7-test

make KERNEL_DIR=/usr/src/linux

make install KERNEL_DIR=/usr/src/linux
```

A questo punto, se non vengono visualizzati errori, possiamo verificare che il supporto ad L7-Filter è effettivamente abilitato richiamando il manuale di Iptables (`man iptables`).



(Figura 3.1 – Man Iptables aggiornato dopo aver abilitato il supporto ad L7-Filter)

3.3 Configurazione Traffic control

Nonostante sia già possibile, a questo punto, inserire in Iptables regole che utilizzano L7-Filter, le attività di Traffic-shaping verranno effettuate da Traffic control, passiamo quindi alla configurazione di quest'ultimo.

La sintassi utilizzata da Traffic control è già stata introdotta nel capitolo 2.4; in questo spiegherò, invece, per cosa è stato utilizzato questo tool.

Si è deciso di suddividere la banda massima in cinque sottoclassi rappresentanti il 75, 50, 25, 10 e 5% del totale; ognuna di queste potrà poi essere caratterizzata da una priorità di instradamento alta o bassa.

Andremo quindi ad implementare undici classi (banda massima, cinque sottoclassi con alta priorità e cinque con bassa priorità), nelle quali i pacchetti verranno *inseriti* in base al numero assegnatogli dal mark effettuato da Netfilter.

Nella classe con banda massima verranno accodati i pacchetti non marcati e quelli relativi alle applicazioni cui non si vuole imporre dei limiti di banda.

A differenza di Iptables, che tramite i tool *iptables-save* ed *iptables-restore*, è predisposto al salvataggio e caricamento delle regole; questo non è possibile per Traffic control; per sopperire a questa mancanza si è quindi scelto di utilizzare uno script che, all'avvio del sistema, configurasse automaticamente Traffic control e, per semplificare la gestione del firewall, anche le regole di Iptables precedentemente salvate.

Inoltre, la modalità con cui Iptables viene gestito non è standard tra le varie distribuzioni Linux; in quelle basate su Red Hat, ad esempio, sono già disponibili stript preconfigurati per la gestione del firewall che, se si vuole, eseguendo automaticamente *iptables-save* ed *iptables-restore* per salvare la configurazione all'arresto del sistema e caricarla di nuovo all'avvio; in altre, come Debian, tali script non sono presenti, nonostante siano presenti i tool per gestire il salvataggio e caricamento delle regole.

L'uso di uno script per la gestione di Iptables e Traffic control ci garantisce quindi una maggiore compatibilità tra le distribuzioni linux.

Un'esempio completo di questo script è riportato nell'Appendice A.

4. L7-Firewall

Una volta introdotti i tool utilizzati in questo progetto (Netfilter e Traffic control) vediamo in che modo l'utente può interagire con il sistema, definire le proprie regole e visualizzare quelle attive.

4.1 Gestione – Il terminale

Tramite shell è possibile inserire nuove regole, cancellare quelle esistenti e visualizzare quali sono attive.

L'uso della shell è consigliabile solo nel caso non si possa utilizzare l'interfaccia web di gestione; la configurazione iniziale del sistema deve comunque essere eseguita tramite l'interfaccia web in quanto è necessario specificare diversi parametri necessari per il funzionamento del firewall.

Ricordiamo che per poter interagire con Iptables è necessario che l'utente abbia eseguito il login sulla macchina autenticandosi come utente root.

Come si è già detto, le funzioni che si voleva implementare per gestire un protocollo sono quelle necessarie a:

- Bloccarlo.
- Assegnargli una banda massima.
- Assegnargli una priorità di instradamento (alta o bassa).

L'inserimento e cancellazione delle regole in Iptables tramite terminale avviene seguendo la sintassi:

```
iptables [-t table] -[AD] chain rule-specification
[options]
```

Come si può ben vedere, il primo parametro da inserire è la tabella a cui si fa riferimento, se questa non viene specificata viene utilizzata quella di default, filter.

Il secondo parametro indica se si sta inserendo una nuova regola (-A) o se si sta cancellando una già esistente (-D).

Il terzo parametro indica la catena (chain) ed infine si specifica l'azione che il firewall dovrà applicare (più eventualmente delle opzioni aggiuntive).

Nel mio caso, il blocco di un protocollo è stato effettuato utilizzando la tebella filter e la catena FORWARD; per marcare i pacchetti di una connessione, invece, si è utilizzata la tabella mangle e la catena PREROUTING.

Volendo bloccare un'applicazione (es. bittorrent) utilizzando il classificatore L7 Filter il comando sarà:

```
iptables –t filter –A FORWARD –m layer7 --l7proto
bittorrent –j DROP
```

In questo caso, il protocollo bit torrent viene bloccato scartandone (drop) i pacchetti relativi.

Per marcare i pacchetti di una connessione, assegnandogli un numero identificativo (es. 10), il comando da inserire sarà:

```
iptables -t mangle -A PREROUTING -i eth2 -m layer7 --
l7proto bittorrent -j MARK --set-mark A
```

Dove A è il numero 10 espresso in base esadecimale.

La cancellare di una regola avviene in maniera analoga all'inserimento, specificando l'opzione –D (delete) al posto di –A (add).

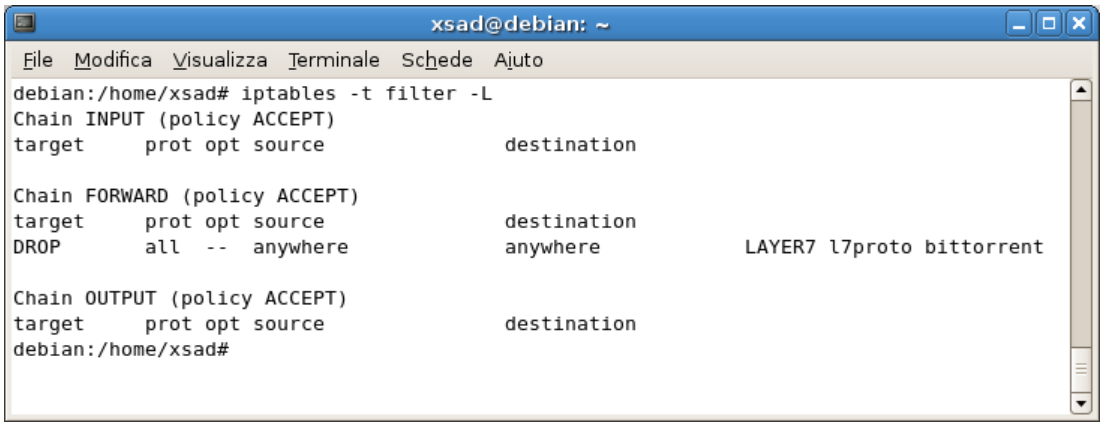
```
iptables -t filter -D FORWARD -m layer7 --l7proto
bittorrent -j DROP

iptables -t mangle -A PREROUTING -i eth2 -m layer7 --
l7proto bittorrent -j MARK --set-mark A
```

È inoltre possibile visualizzare le regole attive tramite il comando

```
iptables [-t table] -L [chain] [options]
```

Volendo, ad esempio, visualizzare le regole attive per la tabella filter, tramite il comando `iptables -t filter -L` otteniamo:



(Figura 3.2 – Visualizzazione regole attive Iptables)

Come si può vedere, attualmente è attiva una regola inserita nella tabella filter, in chain FORWARD che specifica di scartare i pacchetti riconosciuti come appartenenti a bit torrent provenienti o diretti verso qualsiasi host.

Il salvataggio delle regole va effettuato tramite il comando *iptables-save* indicando il path del file di salvataggio predefinito (di default /etc/iptables.conf).

```
iptables-save > /etc/iptables.conf
```

4.2 Gestione – L'interfaccia web

La maggior parte delle distribuzioni Linux dispone già di un tool grafico per facilitare la gestione di Netfilter che però non prevede l'uso del classificatore L7-Filter.

Si è preferito sviluppare un'interfaccia web, piuttosto che un'applicazione da eseguire localmente, per garantire una maggiore versatilità nell'uso della stessa, consentendo quindi di accedervi da qualsiasi terminale dotato di browser web presente nella rete locale.

L'interfaccia web è finalizzata alla gestione delle regole inerenti al classificatore L7-Filter anche se è possibile inserire regole standard.

Si è scelto, per comodità, di far eseguire l'interfaccia web al web application server Apache Tomcat.

Tomcat è disponibile nei repository ufficiali della maggior parte delle distribuzioni linux; la sua installazione è quindi decisamente semplificata.

Le pagine web sono state realizzate in JSP (Java Server Pages) al fine di, eventualmente, poter riutilizzare le classi java implementate nella web application per creare un'applicazione stand-alone per gestire il firewall.

Inserire o modificare le regole in Iptables richiede che l'utente abbia privilegi di amministratore, per gestire questa situazione è stata utilizzata una libreria open source distribuita dalla Trilead (http://www.trilead.com) che consente di implementare semplicemente una connessione ssh in java.

L'uso di questa libreria, ha permesso di eseguire l'applicazione con bassi privilegi nell'uso normale; aumentandoli solo quando necessario effettuando una connessione ssh con i dati di autenticazione forniti e chiudendo la connessione dopo aver eseguito i comandi.

La prima pagina è la schermata di login, per accedere al sistema l'utente deve autenticarsi come root.

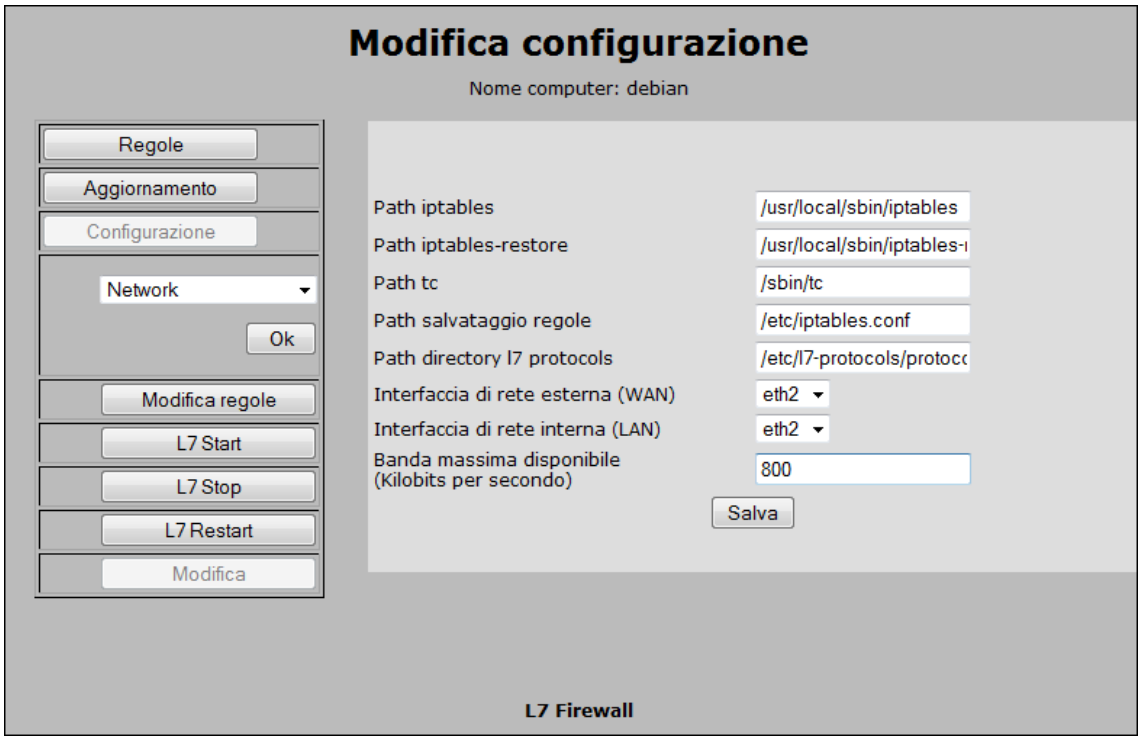
(Figura 3.3 – Pagina di login)

Se il controllo sui dati inseriti da esito negativo viene riproposta la schermata di login, altrimenti, se l'autenticazione ha dato esito positivo:

- nel caso non sia presente il file di configurazione (es. al primo avvio del sistema) si passa alla pagina di setup.
- nel caso sia presente il file di configurazione si passa alla pagina principale.

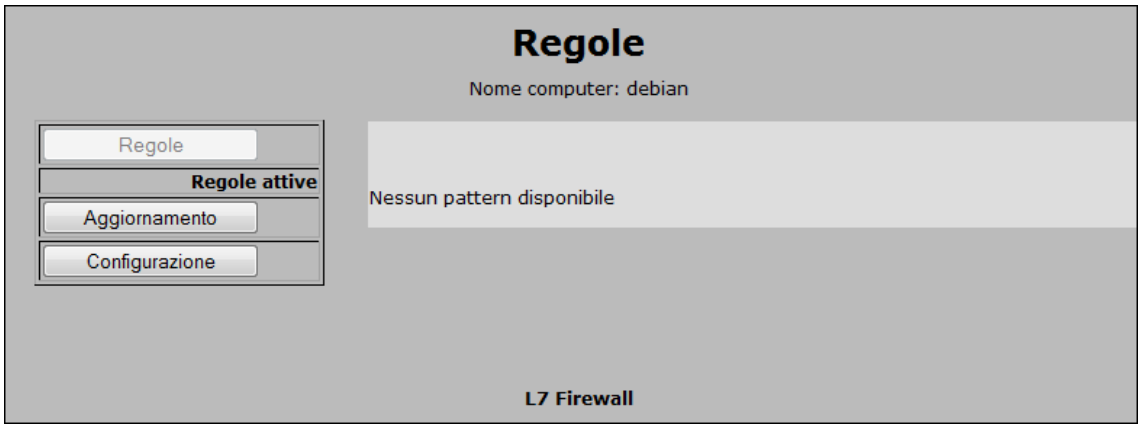
Nella pagina di setup vanno specificati:

- i path dei file binari utilizzati (iptables, iptables-restore e traffic control).
- Il path del file di salvataggio delle regole.
- Il path della directory contenente i pattern.
- Il nome delle interfacce di rete.
- La banda massima disponibile.



(Figura 3.4 – Pagina di setup)

Premendo il tasto salva viene creato lo script di avvio in /etc/init.d e si apre la pagina principale dalla quale si gestiscono le regole relative ad L7-Filter.



(Figura 3.5 – Pagina principale 1)

Al primo avvio, non essendoci pattern installati, bisogna passare alla pagina Aggiornamento per scaricarne la versione più recente.

(Figura 3.6 – Pagina aggiornamento 1)

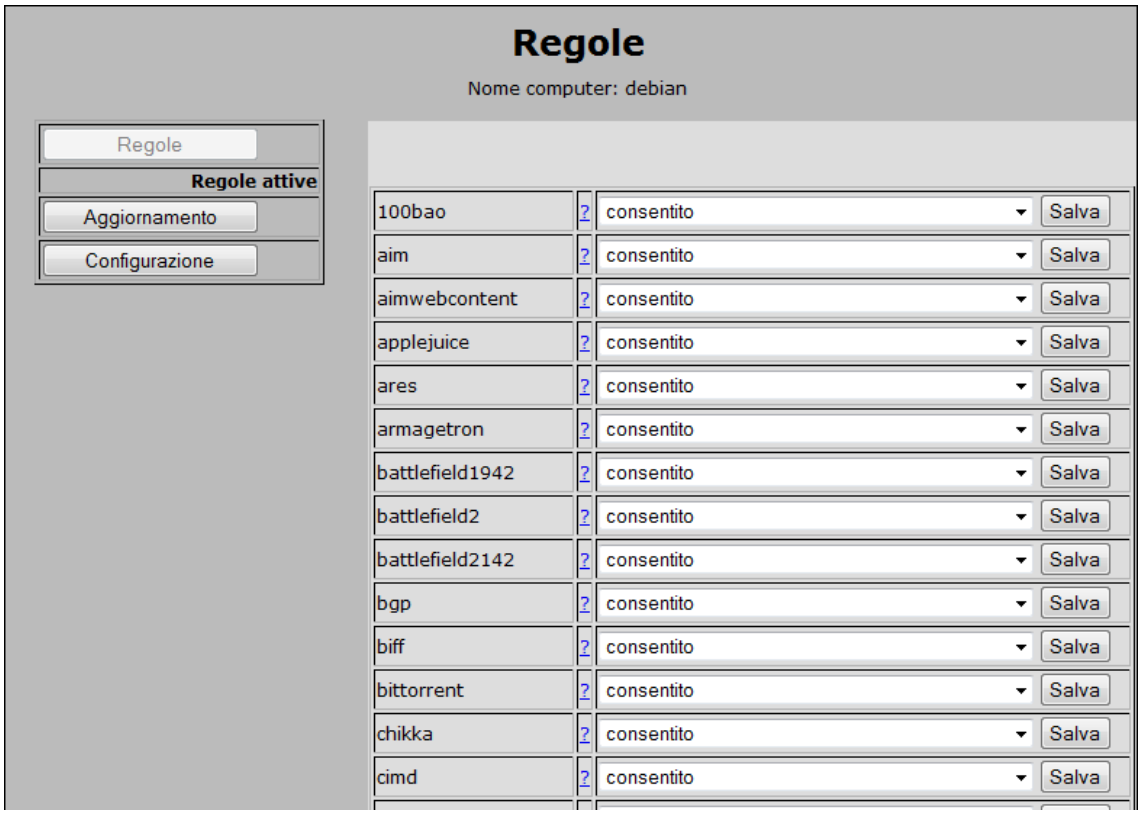
Il link punta alla pagina di download ufficiale, da li si copia l'url del pacchetto contenente i pattern nell'apposita casella ed il sistema provvederà a scaricarlo ed installarlo.

Come si può vedere nella figura 3.6 il campo "Versione attuale" è vuoto, non essendoci pattern installati.

(Figura 3.7 - Pagina aggiornamento 2)

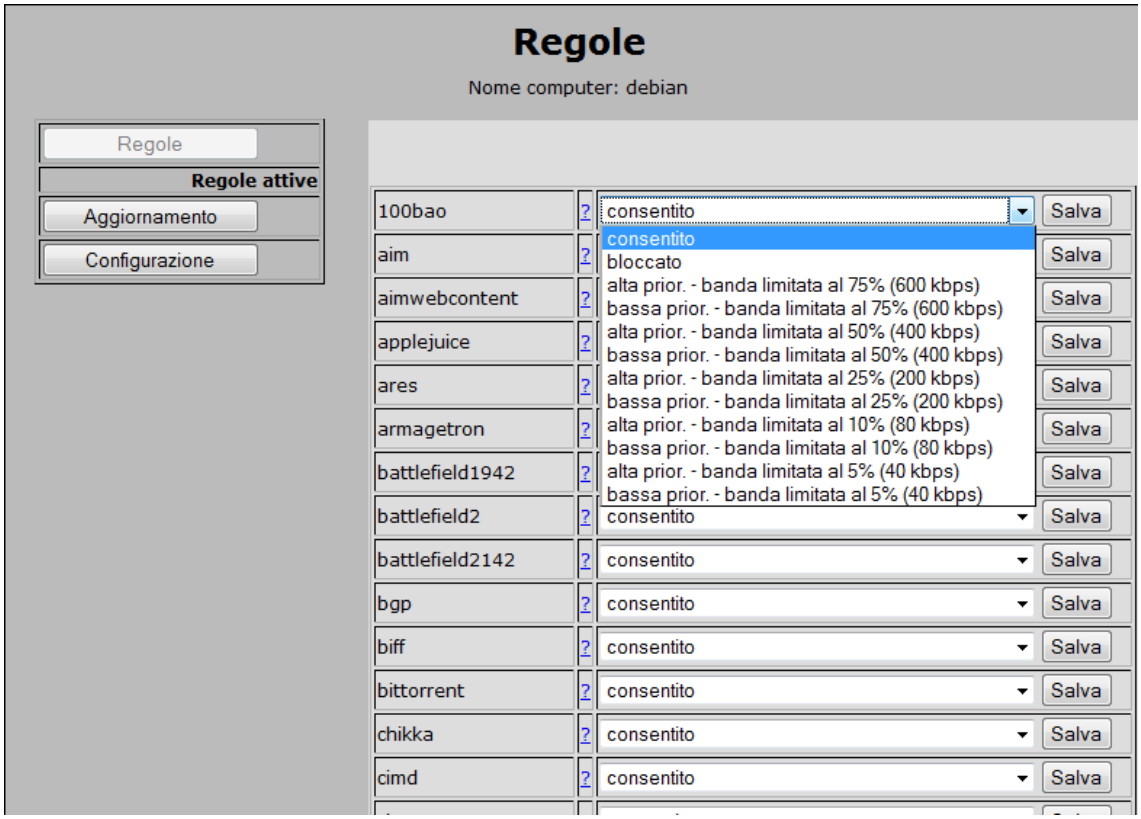
Ad operazione ultimata il campo "Versione attuale" viene aggiornato.

A questo punto, tornando alla pagina principale è possibile vedere i protocolli riconosciuti e la relativa regola attiva.



(Figura 3.8 – Pagina principale 2)

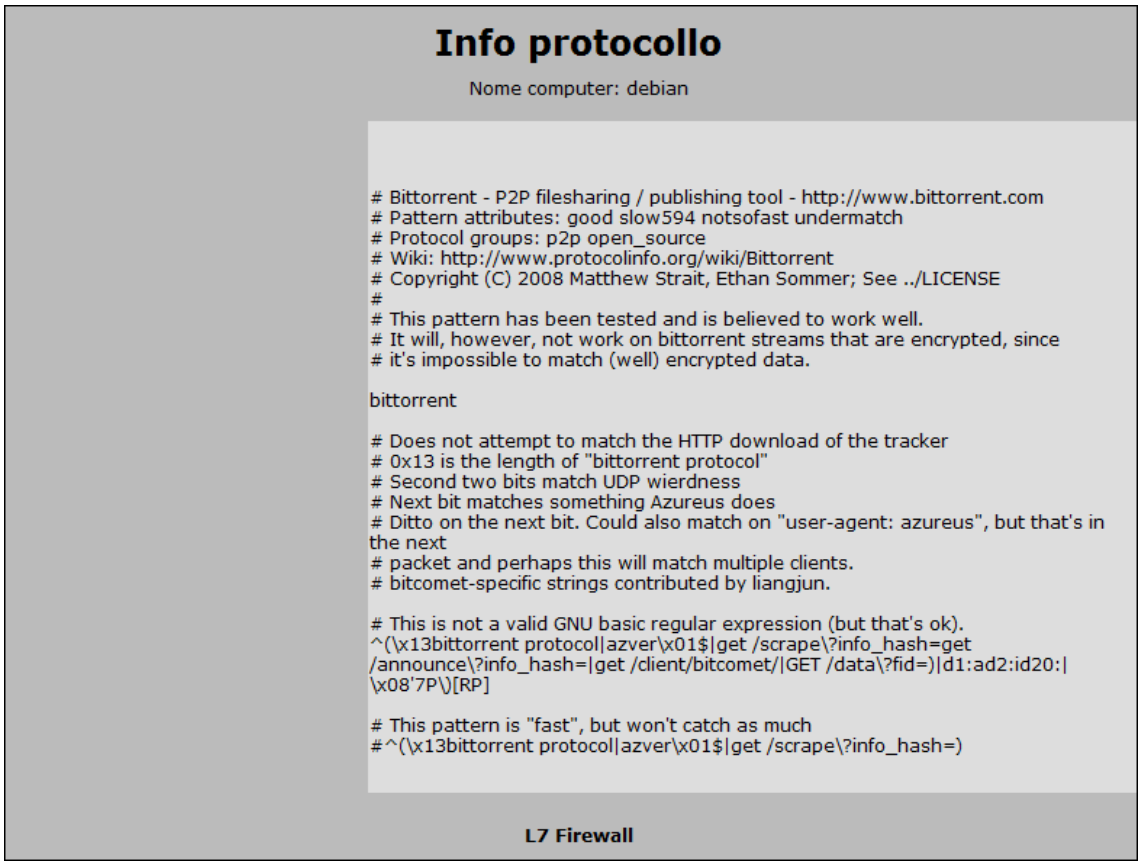
Dai menù a discesa è possibile impostare la modalità con cui il protocollo deve essere gestito (consentendolo, bloccandolo, limitandone la banda disponibile ed assegnandogli una priorità di instradamento).



(Figura 3.9 – Regole applicabili)

I valori di banda massima che è possibile impostare sono stati calcolati in base al valore impostato nella pagina di configurazione.

I collegamenti “?”, a fianco di oggi voce, visualizzano la descrizione del protocollo letta dal pattern relativo al protocollo scelto.



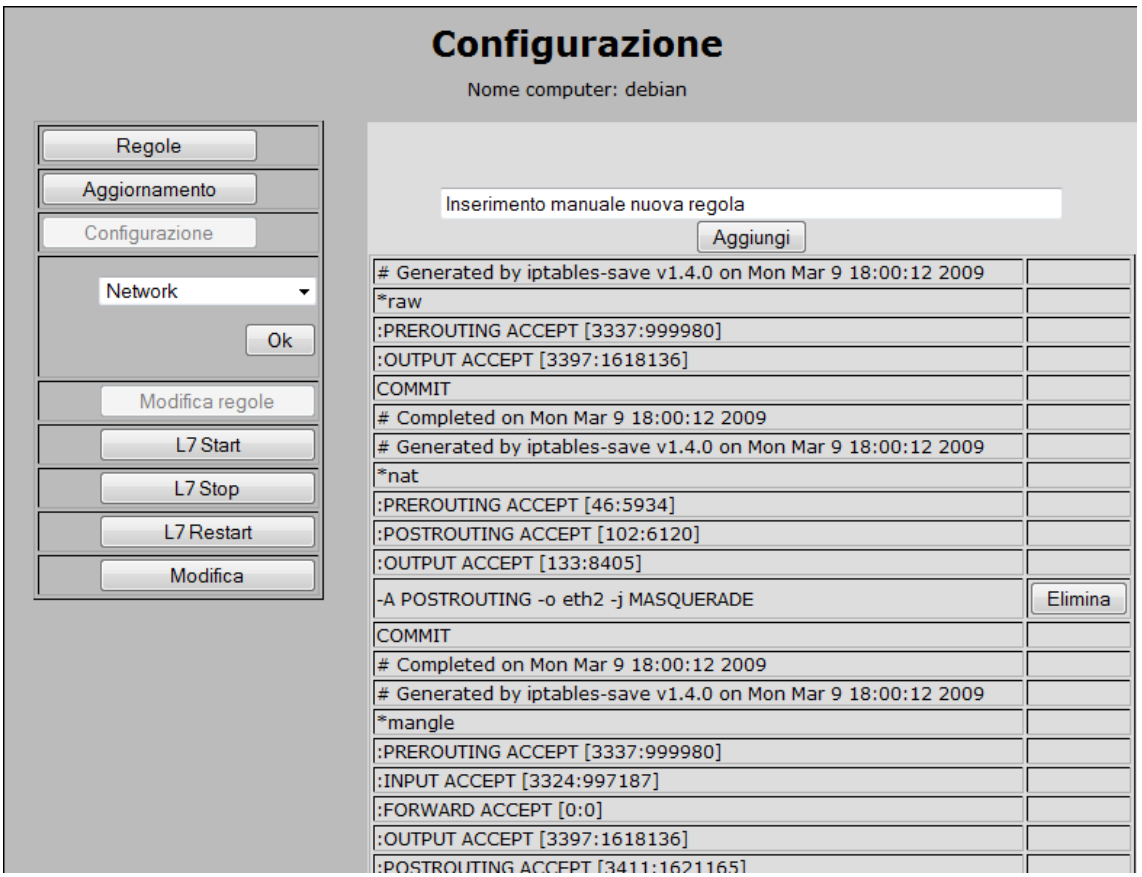
(Figura 3.10 – Pagina info protocollo)

Dalla pagina “Configurazione” è possibile:

- Visualizzare:
  - Le interfacce di rete e la relativa configurazione.
  - Il script di avvio.
  - Il file di salvataggio delle regole.
  - I manuali man di Iptables e Traffic control.
  - Lo stato del firewall.

(Figura 3.11 – Pagina configurazione)

- Inserire e cancellare manualmente le regole di iptables (non solo quelle relative ad L7-Filter)



(Figura 3.12 – Pagina setup manuale regole Iptables)

- Eseguire le opzioni start, stop e restart dello script di avvio.
- Modificare il file di configurazione.

5 CONCLUSIONI

È stato realizzato un sistema che permette di definire politiche di traffic-shaping, basando il progetto sull'utilizzo di strumenti open-source o comunque gratuiti.

Lo scopo principale del progetto era l'implementazione di un firewall non tanto indirizzato alla gestione della sicurezza (esistono già innumerevoli strumenti atti a questo) quanto alla gestione del traffico utilizzabile dalle varie applicazioni che gli utenti connessi alla rete potrebbero utilizzare, per impedire l'uso improprio della rete e, al tempo stesso, garantire un funzionamento ottimale per le applicazioni che si desidera privilegiare.

I test effettuati per verificare il corretto funzionamento del firewall sono stati eseguiti principalmente tramite il software VMware, utilizzando cinque macchine virtuali per simulare gli host ed un’altra per eseguire il firewall.

In tutte le prove effettuate, il funzionamento del firewall ha soddisfatto ampiamente le aspettative, non è stato però possibile eseguire test più approfonditi in una rete reale con un numero maggiore di host, al fine di valutare la stabilità del sistema in condizioni di carico di lavoro più alto.

L7-Filter si è rivelato un ottimo classificatore, potente ed intuitivo; gli unici appunti che mi sento di fare riguardano la necessità di compilazione del kernel e di iptables che, di fatto, limitano le possibilità di aggiornamento per il sistema operativo; inoltre, come evidenziato anche dalla documentazione ufficiale, l'uso di espressioni regolari per effettuare il match con i pacchetti di dati può causare riscontri errati nell'individuazione dell'applicazione che li ha generati (cosa che, però, finora non sono riuscito a riscontare).

6 BIBLIOGRAFIA

- Gheorghe Lucian, **Designing and Implementing Linux Firewalls with QoS using netfilter, iproute2, NAT and L7-filter**, Packt Publishing (October 31, 2006)

7 APPENDICE A – LO SCRIPT DI AVVIO

Lo script viene creato al primo avvio dell'interfaccia web; le prime righe specificano i path di:

- File binari utilizzati (iptables, iptables-restore e traffic control).
- File di salvataggio delle regole.
- Directory contenente i pattern.
- Nome delle interfacce di rete.
- Banda massima (espressa in kilobit per secondo).
- Versione dell'archivio dei pattern.

Le opzioni definite sono:

- Start.
- Stop.
- Restart.
- Status.

Start viene eseguito automaticamente all'avvio della macchina e si occupa inizialmente di verificare la presenza del file di salvataggio delle regole (creandolo altrimenti), successivamente abilita il masquerading, carica le regole salvate di iptables tramite il comando `iptables-restore` ed infine crea le qdisc necessarie per traffic control.

Stop viene utilizzato per simulare lo spegnimento del firewall, nello specifico vengono cancellate le regole attive di iptables e le qdisc. Essendo tali regole salvate su file, queste possono essere comunque ricaricate tramite il comando start o restart.

Restart esegue il comando stop e successivamente start.

Status ci consente di visualizzare le regole attive di iptables relative alle tabelle mangle e filter.

Un esempio di script di avvio:

```
#!/bin/bash

#

# L7 Firewall startup script



IPTABLES=/usr/local/sbin/iptables

IPTABLES_RESTORE=/usr/local/sbin/iptables-restore

TC=/sbin/tc

EXT_IFACE=eth2

INT_IFACE=eth3

PATH_IPTABLES_SAVE=/etc/iptables.conf

PATH_PROTOCOLLI=/etc/l7-protocols/protocols/

BAND_MAX100=800

PROTO_VERSION=l7-protocols-2008-12-18




case "$1" in

start)

echo "Starting l7 firewall"

touch $PATH_IPTABLES_SAVE
```

```
$IPTABLES_RESTORE < $PATH_IPTABLES_SAVE

$IPTABLES -t nat -F POSTROUTING

$IPTABLES -t nat -A POSTROUTING -o $EXT_IFACE -j
MASQUERADE

$TC qdisc del dev $INT_IFACE root

$TC qdisc add dev $INT_IFACE root handle 1: htb default
10

$TC class add dev $INT_IFACE parent 1: classid 1:1 htb
rate 720kbit ceil 800kbit prio 1

$TC class add dev $INT_IFACE parent 1:1 classid 1:10 htb
rate 720kbit ceil 800kbit prio 1

$TC class add dev $INT_IFACE parent 1:1 classid 1:11 htb
rate 540kbit ceil 600kbit prio 1

$TC class add dev $INT_IFACE parent 1:1 classid 1:12 htb
rate 360kbit ceil 400kbit prio 1

$TC class add dev $INT_IFACE parent 1:1 classid 1:13 htb
rate 180kbit ceil 200kbit prio 1

$TC class add dev $INT_IFACE parent 1:1 classid 1:14 htb
rate 1kbit ceil 80kbit prio 1

$TC class add dev $INT_IFACE parent 1:1 classid 1:15 htb
rate 1kbit ceil 40kbit prio 1

$TC class add dev $INT_IFACE parent 1:1 classid 1:16 htb
rate 540kbit ceil 600kbit prio 2

$TC class add dev $INT_IFACE parent 1:1 classid 1:17 htb
rate 360kbit ceil 400kbit prio 2
```

```
$TC class add dev $INT_IFACE parent 1:1 classid 1:18 htb
rate 180kbit ceil 200kbit prio 2

$TC class add dev $INT_IFACE parent 1:1 classid 1:19 htb
rate 1kbit ceil 80kbit prio 2

$TC class add dev $INT_IFACE parent 1:1 classid 1:20 htb
rate 1kbit ceil 40kbit prio 2

$TC filter add dev $INT_IFACE protocol ip parent 1:0
handle 100 fw flowid 1:10

$TC filter add dev $INT_IFACE protocol ip parent 1:0
handle 175 fw flowid 1:11

$TC filter add dev $INT_IFACE protocol ip parent 1:0
handle 150 fw flowid 1:12

$TC filter add dev $INT_IFACE protocol ip parent 1:0
handle 125 fw flowid 1:13

$TC filter add dev $INT_IFACE protocol ip parent 1:0
handle 110 fw flowid 1:14

$TC filter add dev $INT_IFACE protocol ip parent 1:0
handle 105 fw flowid 1:15

$TC filter add dev $INT_IFACE protocol ip parent 1:0
handle 275 fw flowid 1:16

$TC filter add dev $INT_IFACE protocol ip parent 1:0
handle 250 fw flowid 1:17

$TC filter add dev $INT_IFACE protocol ip parent 1:0
handle 225 fw flowid 1:18
```

```
$TC filter add dev $INT_IFACE protocol ip parent 1:0
handle 210 fw flowid 1:19

$TC filter add dev $INT_IFACE protocol ip parent 1:0
handle 205 fw flowid 1:20

$TC qdisc del dev $EXT_IFACE root

$TC qdisc add dev $EXT_IFACE root handle 1: htb default
10

$TC class add dev $EXT_IFACE parent 1: classid 1:1 htb
rate 720kbit ceil 800kbit prio 1

$TC class add dev $EXT_IFACE parent 1:1 classid 1:10 htb
rate 720kbit ceil 800kbit prio 1

$TC class add dev $EXT_IFACE parent 1:1 classid 1:11 htb
rate 540kbit ceil 600kbit prio 1

$TC class add dev $EXT_IFACE parent 1:1 classid 1:12 htb
rate 360kbit ceil 400kbit prio 1

$TC class add dev $EXT_IFACE parent 1:1 classid 1:13 htb
rate 180kbit ceil 200kbit prio 1

$TC class add dev $EXT_IFACE parent 1:1 classid 1:14 htb
rate 1kbit ceil 80kbit prio 1

$TC class add dev $EXT_IFACE parent 1:1 classid 1:15 htb
rate 1kbit ceil 40kbit prio 1

$TC class add dev $EXT_IFACE parent 1:1 classid 1:16 htb
rate 540kbit ceil 600kbit prio 2

$TC class add dev $EXT_IFACE parent 1:1 classid 1:17 htb
rate 360kbit ceil 400kbit prio 2
```

```
$TC class add dev $EXT_IFACE parent 1:1 classid 1:18 htb
rate 180kbit ceil 200kbit prio 2

$TC class add dev $EXT_IFACE parent 1:1 classid 1:19 htb
rate 1kbit ceil 80kbit prio 2

$TC class add dev $EXT_IFACE parent 1:1 classid 1:20 htb
rate 1kbit ceil 40kbit prio 2

$TC filter add dev $EXT_IFACE protocol ip parent 1:0
handle 100 fw flowid 1:10

$TC filter add dev $EXT_IFACE protocol ip parent 1:0
handle 175 fw flowid 1:11

$TC filter add dev $EXT_IFACE protocol ip parent 1:0
handle 150 fw flowid 1:12

$TC filter add dev $EXT_IFACE protocol ip parent 1:0
handle 125 fw flowid 1:13

$TC filter add dev $EXT_IFACE protocol ip parent 1:0
handle 110 fw flowid 1:14

$TC filter add dev $EXT_IFACE protocol ip parent 1:0
handle 105 fw flowid 1:15

$TC filter add dev $EXT_IFACE protocol ip parent 1:0
handle 275 fw flowid 1:16

$TC filter add dev $EXT_IFACE protocol ip parent 1:0
handle 250 fw flowid 1:17

$TC filter add dev $EXT_IFACE protocol ip parent 1:0
handle 225 fw flowid 1:18
```

```
$TC filter add dev $EXT_IFACE protocol ip parent 1:0
handle 210 fw flowid 1:19

$TC filter add dev $EXT_IFACE protocol ip parent 1:0
handle 205 fw flowid 1:20

;;

stop)

echo "Stopping l7 firewall"

$IPTABLES -t mangle -F PREROUTING

$IPTABLES -t filter -F FORWARD

$TC qdisc del dev $INT_IFACE root

$TC qdisc del dev $EXT_IFACE root

;;


restart)

echo "Restarting l7 firewall"

$0 stop

$0 start

;;


status)

echo "Status"

echo ""
```

```
echo ""

echo ""

echo "MANGLE"

echo ""

echo ""

echo ""

$IPTABLES -t mangle -L

echo ""

echo ""

echo ""

echo "FILTER"

echo ""

echo ""

echo ""

$IPTABLES -t filter -L

;;



*) echo "Usage: /etc/init.d/l7Config
{start|stop|restart|status}"

exit 2

;;
```

```
esac

exit 0
```