Bandwidth Limiting HOWTO

Tomasz Chmielewski

tch@metalab.unc.edu

Questo documento descrive come impostare un server Linux in modo da limitare la banda passante in entrata o in uscita e come usare la connessione internet in modo più efficiente.

Traduzione di Emanuele Tajariol - Maggio 2002.

Sommario

1. Introduzione 3

1.1. Versioni recenti di questo documento 3
1.2. Liberatoria 3
1.3. Copyright e licenza d'uso 3
1.4. Feedback e correzioni 3
1.5. Ringraziamenti 3

2. Prima di iniziare 3

2.1. Cosa serve 4
2.2. Come funziona? 4

3. Installazione e configurazione del software necessario 4

3.1. Installazione di Squid con l'opzione "delay pools" 5
3.2. Configurazione di Squid per usare i delay pool 5
3.3. Risoluzione dei problemi rimanenti 9

4. Occuparsi di altri protocolli ad alto traffico usando CBQ 10

4.1. FTP 11
4.2. Napster, Realaudio, Windows Media e altre questioni 11

5. FAQ 12

5.1. È possibile limitare la banda utente per utente con i delay pool? 13
5.2. Come posso far funzionare **wget** con Squid? 13
5.3. Ho impostato il mio server SOCKS in ascolto sulla porta 1080 ed ora non riesco più a collegarmi ad alcun server IRC. 13
5.4. Non voglio che Kazaa o Audiogalaxy occupino tutta la mia banda di ingresso. 13
5.5. Il mio server di posta in uscita consuma tutta la banda. 14
5.6. Posso limitare il mio server FTP o WWW in un modo simile a quello mostrato nella domanda precedente? 14
5.7. È possibile limitare la banda utente per utente con lo script **cbq.init**? 14
5.8. Ogni volta che lancio `cbq.init`, dice che non trova `sch_cbq`. 14

5.9. CBQ a volte non funziona senza alcun motivo. 14
5.10. I delay pool sono stupidi; perché non posso scaricare qualcosa a piena velocità quando sono l'unico ad usare la rete? 15
5.11. I miei download si bloccano alle 23:59 per colpa di "acl day time 09:00-23:59" in squid.conf. Posso farci qualcosa?....................15
5.12. I log di Squid continuano a crescere molto velocemente; posso risolvere in qualche modo? 15
5.13. CBQ è stupido: perché non posso scaricare qualcosa a piena velocità quando sono l'unico ad usare la rete?....................16
6. Varie 17
6.1. Risorse utili 17

1. Introduzione

Lo scopo di questa guida è fornire una semplice soluzione per limitare il traffico in ingresso, evitando così che gli utenti della LAN esauriscano la banda passante della connessione internet.

Ciò è utile quando la connessione internet è lenta, oppure quando gli utenti della rete locale scaricano una notevole quantità di mp3 e le immagini ISO delle più recenti distribuzioni Linux.

1.1. Versioni recenti di questo documento

Si può leggere la versione più recente di questo documento sul WWW presso http://www.tldp.org.

Nuove versioni di questo documento saranno presenti anche sui vari siti WWW e FTP riguardanti Linux, compresa la home page di LDP (http://www.tldp.org).

1.2. Liberatoria

Né l'autore né i distributori, o qualsiasi altro collaboratore di questo HOWTO, sono in qualsiasi modo responsabili per danni fisici, economici, morali o di qualsiasi altra natura subiti seguendo le indicazioni di questo testo.

Liberatoria originale: [Neither the author nor the distributors, or any other contributor of this HOWTO are in any way responsible for physical, financial, moral or any other type of damage incurred by following the suggestions in this text.]

1.3. Copyright e licenza d'uso

1.4. Feedback e correzioni

Se si hanno domande o commenti su questo documento, si prega di inviare una email a Tomasz Chmielewski all'indirizzo *tch@metalab.unc.edu* (mailto:tch@metalab.unc.edu). Suggerimenti e critiche saranno ben accetti. Se trovate errori (ce ne saranno parecchi dato che l'inglese non è la mia lingua nativa), anche di battitura, fatemelo sapere in modo da poterli correggere nella prossima versione. Grazie.

1.5. Ringraziamenti

Vorrei ringraziare Ami M. Echeverri (lula@pollywog.com) che mi ha aiutato a convertire questo HOWTO nel formato SGML ed ha corretto alcuni errori. Vorrei anche ringraziare Ryszard Prosowicz (prosowicz@poczta.fm) per gli utili suggerimenti.

2. Prima di iniziare

Si immagini la seguente situazione:

- Una connessione internet punto-punto (ppp) a 115,2 Kbit/s (via modem) (115,2/10 = 11,5 Kbyte/s). Nota: con connessioni ethernet (via scheda di rete) si deve dividere 115,2 per 8; con ppp si deve dividere per 10, a causa dei bit di start e stop (8 + 1 + 1 = 10).
- Alcune postazioni della LAN e relativi utenti continuano ad effettuare download voluminosi.
- Le pagine web devono essere visualizzate velocemente, a prescindere da quanti download si stiano effettuando.
- L'interfaccia verso internet è **ppp0**.
- L'interfaccia LAN è **eth0**.
- L'indirizzo di rete locale è 192.168.1.0/24.

2.1. Cosa serve

Ci si creda o no, il traffic shaping è facile da ottenere e non si deve per forza leggere una tonnellata di libri su algoritmi di routing o di gestione delle code.

Affinché tutto funzioni, serve almeno il proxy Squid; se si vuole metterlo a punto in modo preciso si deve familiarizzare con ipchains o iptables e con CBQ.

Per testare il lavoro fatto si può installare IPTraf.

2.2. Come funziona?

Squid è probabilmente il più avanzato proxy server HTTP disponibile per Linux. Può essere d'aiuto a risparmiare larghezza di banda in due modi:

- La prima e fondamentale caratteristica di un proxy server: mantiene in memoria o su disco le pagine web, le immagini e gli altri oggetti già scaricati. Così se due persone richiedono la stessa pagina web questa non viene scaricata da internet ma dal proxy locale.
- Oltre il caching normale, Squid ha una funzionalità particolare chiamata "delay pools" ("gruppi di collegamenti rallentati"). Grazie ai delay pool è possibile limitare il traffico internet in un modo ragionevole, a seconda delle cosiddette "parole magiche" presenti in ogni URL. Ad esempio una parola magica potrebbe essere ".mp3", ".exe", ".avi" ecc. Ogni parte distinta di un URL (tipo .avi) può essere definita come parola magica.

Con ciò si può dire a Squid di scaricare questi tipi di file ad una velocità specifica (nel nostro esempio questa sarà di circa 5 Kbyte/s). Se gli utenti della LAN scaricano contemporaneamente questi tipi di file, questi saranno scaricati ad una velocità complessiva di circa 5 Kbyte/s, lasciando la banda restante per le pagine web, per la posta elettronica, le news, irc, ecc...

Chiaramente internet non è usato solamente per scaricare file via pagine web (tramite http o ftp). In seguito verrà illustrato come limitare la banda per Napster, Realaudio e altre possibilità.

3. Installazione e configurazione del software necessario

In questo capitolo si spiega come installare il software necessario per limitare e controllare l'uso di banda.

3.1. Installazione di Squid con l'opzione "delay pools"

Come detto precedentemente, Squid ha una funzionalità chiamata "delay pools", che permette di controllare la banda disponibile per i download. Purtroppo nella maggior parte delle distribuzioni Squid viene fornito senza questa opzione.

Quindi se Squid è già installato c'è una una brutta notizia: si dovrà disinstallarlo e poi reinstallarlo nuovamente, abilitando i delay pool come illustrato più avanti.

1. Per ottenere prestazioni ottimali dal proxy Squid è preferibile creare una partizione separata per la sua cache, chiamata /cache/; la sua dimensione dovrebbe essere all'incirca di 300 megabyte, variabile secondo i propri bisogni.

   In caso non si sappia creare una partizione separata, si può creare la directory /cache/ su una partizione principale, ma le prestazioni di Squid ne risentiranno un po'.

2. Si crei un utente "squid" sicuro:

   ```
   # useradd -d /cache/ -r -s /dev/null squid >/dev/null 2>&1
   ```

   Nessuno potrà fare login come squid, nemmeno l'utente root.

3. Si scarichino i sorgenti di Squid da http://www.squid-cache.org.

   Al momento della stesura, la versione più recente era Squid 2.4 stable1:

   http://www.squid-cache.org/Versions/v2/2.4/squid-2.4.STABLE1-src.tar.gz

4. Si estragga il tutto in `/var/tmp`:
5. `# tar xzpf squid-2.4.STABLE1-src.tar.gz`
6. Si compili e si installi Squid (scrivere tutto su un'unica riga):

   ```
   # ./configure --prefix=/opt/squid --exec-prefix=/opt/squid --enable-delay-pools
   --enable-cache-digests --enable-poll --disable-ident-lookups --enable-truncate
   --enable-removal-policies
   ```

   ```
   # make all
   ```

   ```
   # make install
   ```

3.2. Configurazione di Squid per usare i delay pool

1. Si configuri il file squid.conf (situato in /opt/squid/etc/squid.conf):

```
#squid.conf
# Ogni opzione in questo file è molto ben documentata nel file squid.conf originale
# e su http://www.visolve.com/squidman/Configuration%20Guide.html
#
# Le porte su cui Squid si porrà in ascolto
http_port 8080
icp_port 3130
# per i cgi-bin non verrà usata la cache
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
# Quantità di memoria usata da Squid. Beh, Squid ne userà ben di più.
cache_mem 16 MB
# 250 significa che Squid userà 250 megabyte di spazio su disco.
cache_dir ufs /cache 250 16 256

# File dove Squid salverà i log.
cache_log /var/log/squid/cache.log
cache_access_log /var/log/squid/access.log
cache_store_log /var/log/squid/store.log
cache_swap_log /var/log/squid/swap.log
# Periodo di rotazione dei file prima di essere cancellati.
# Leggere le FAQ per ulteriori informazioni.
logfile_rotate 10

redirect_rewrites_host_header off
cache_replacement_policy GDSF
acl localnet src 192.168.1.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255
acl Safe_ports port 80 443 210 119 70 20 21 1025-65535
acl CONNECT method CONNECT
acl all src 0.0.0.0/0.0.0.0
http_access allow localnet
http_access allow localhost
http_access deny !Safe_ports
http_access deny CONNECT
http_access deny all
maximum_object_size 3000 KB
store_avg_object_size 50 KB

# Impostare queste voci se si vuole che il proxy funzioni in modalità trasparente.
# Avere un proxy trasparente significa che generalmente non si dovranno
# configurare i browser di tutti i client, ma ci sono anche alcuni
# inconvenienti.
# Lasciare queste righe decommentate non darà problemi.
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on

# Tutti gli utenti della rete locale saranno visti dai server web esterni
# come se usassero tutti Mozilla per Linux. :)
anonymize_headers deny User-Agent
fake_user_agent Mozilla/5.0 (X11; U; Linux i686; en-US; rv:0.9.6+) Gecko/20011122
```

```
# Per rendere la connessione ancora più veloce, si scrivano due righe simili
# a quelle qui sotto. Punteranno ad un proxy server di gerarchia superiore
# che sarà usato dal nostro Squid. Ci si ricordi di impostare come server quello che
# risulta più veloce!
# Si misurino i ping, i traceroute e tutto il resto.
# Ci si assicuri che le porte http e icp siano corrette.

# Si tolgano i segni di commento (#) dalle righe che iniziano con "cache_peer" se
# necessario. Questo è il proxy che verrà usato per tutte le connessioni...
#cache_peer w3cache.icm.edu.pl parent 8080 3130 no-digest default

# ...tranne per le connessioni ad indirizzi ed IP che iniziano con "!".
# E` una buona idea non usare uno superiore
#cache_peer_domain w3cache.icm.edu.pl !.pl !7thguard.net !192.168.1.1

# Questo è comodo quando si vuole usare il Cache Manager.
# Si copi cachemgr.cgi nella directory cgi-bin del proprio server web.
# Lo si può raggiungere via browser digitando
# l'indirizzo http://il-proprio-server-web/cgi-bin/cachemgr.cgi
cache_mgr propria@email
cachemgr_passwd secret_password all

# Questo è il nome dell'utente con cui Squid si identificherà.
cache_effective_user squid
cache_effective_group squid

log_icp_queries off
buffered_logs on


##### DELAY POOLS
# Questa è la parte più importante per il traffic shaping con Squid.
# Per una descrizione dettagliata si consulti il file squid.conf o la
# documentazione su http://www.squid-cache.org

# Non si vogliono limitare i download sulla rete locale.
acl magic_words1 url_regex -i 192.168

# Si vuole limitare i download di questi tipi di file.
# Tutto questo va su una sola riga.
acl magic_words2 url_regex -i ftp .exe .mp3 .vqf .tar.gz .gz .rpm .zip .rar .avi .mpeg .mpe .mpg .qt
.ram .rm .iso .raw .wav .mov
# Non sono compresi .html, .gif, .jpg e simili poiche' in genere non occupano
# una banda eccessiva.

# Si vuole limitare la banda durante il giorno e permettere una piena
# occupazione di banda durante la notte.
# Attenzione! Con la riga acl qui sotto probabilmente i download si
# interromperanno alle 23:59. Si leggano le FAQ in questo testo se si vuole evitare
# questo tipo di comportamento.
acl day time 09:00-23:59
```

```
# Ci sono due delay_pools diversi
# Si legga la documentazione di Squid per familiarizzare con
# delay_pools e delay_class.
delay_pools 2

# Primo delay pool
# Non si vuole ritardare il traffico locale.
# Ci sono tre classi di delay pool; qui ci si limiterà ad usare solo la seconda.
# Prima classe di delay(1) del secondo tipo(2).
delay_class 1 2

#-1/-1 significa che non ci sono restrizioni.
delay_parameters 1 -1/-1 -1/-1

#la parola magica magic_words1, pari a 192.168, definita in precedenza
delay_access 1 allow magic_words1


# Secondo delay pool.
# Si vuole rallentare il download dei file menzionati in magic_words2.
# Seconda classe di delay (2) del secondo tipo (2).
delay_class 2 2

# I numeri seguenti sono valori in byte;
# ci si ricordi che Squid non considera i bit di start e di stop
# 5000/150000 sono i valori per l'intera rete
# 5000/120000 sono i valori per il singolo IP
# dopo che i file scaricati superano i 150000 byte,
# (o anche due o tre volte questo valore)
# continueranno ad essere scaricati a circa 5000 byte/s
delay_parameters 2 5000/150000 5000/120000

# "day" è stato impostato precedentemente come l'intervallo tra le 09:00 e le 23:59.
delay_access 2 allow day
delay_access 2 deny !day
delay_access 2 allow magic_words2


#EOF
```

Una volta configurato il tutto, ci si deve assicurare che tutto ciò che si trova sotto le directory `/opt/squid` e `/cache` appartenga all'utente "squid".

 # mkdir /var/log/squid/

 # chown squid:squid /var/log/squid/

 # chmod 770 /var/log/squid/

 # chown -R squid:squid /opt/squid/

 # chown -R squid:squid /cache/

Adesso tutto è pronto per lanciare Squid. Quando lo si fa per la prima volta, si devono creare le sue directory di cache:

 # /opt/squid/bin/squid -z

Si lanci Squid, controllando che tutto funzioni correttamente. Un buono strumento per la verifica è IPTraf; lo si può trovare su http://freshmeat.net. Ci si assicuri di aver impostato correttamente il proxy sui browser dei client (192.168.1.1 porta 8080 nel nostro esempio):

 # /opt/squid/bin/squid

Se tutto funziona, si aggiunga la riga `/opt/squid/bin/squid` alla fine degli script di inizializzazione. Di solito è `/etc/rc.d/rc.local`.

Altre utili opzioni di Squid potrebbero essere:

 # /opt/squid/bin/squid -k reconfigure (riconfigura Squid se si è modificato il file squid.conf)

 # /opt/squid/bin/squid -help :) autoesplicativo

Si potrebbe anche copiare `cachemgr.cgi` nella directory cgi-bin del proprio server WWW, per utilizzare un utile Cache Manager.

3.3. Risoluzione dei problemi rimanenti

Bene, è stato installato Squid ed è stato configurato per fargli usare i delay pool. Scommetto che nessuno vuole essere vincolato, in special modo gli utenti ingegnosi della nostra LAN. Probabilmente cercheranno di evitare i limiti che gli abbiamo imposto solo per scaricare i loro mp3 preferiti un po' più velocemente (causandoci così un bel po' di mal di testa).

Suppongo che sulla nostra rete si starà usando l'IP masquerading, in modo che gli utenti possano usare IRC, ICQ, e-mail, ecc. Questo va bene, ma si deve essere sicuri che gli utenti della LAN usino il nostro Squid con delay pool per accedere alle pagine web ed usino `ftp`.

La maggior parte di questi problemi può essere risolto usando `ipchains` (Linux con kernel 2.2.x) o `iptables` (Linux con kernel 2.4.x).

3.3.1. Linux con kernel 2.2.x (ipchains)

Ci si assicuri che nessuno bari usando un proxy che non sia il nostro. Proxy pubblici di solito usano le porte 3128 e 8080:

/sbin/ipchains -A input -s 192.168.1.1/24 -d ! 192.168.1.1 3128 -p TCP -j REJECT

/sbin/ipchains -A input -s 192.168.1.1/24 -d ! 192.168.1.1 8080 -p TCP -j REJECT

Si dovrà anche fare in modo che nessuno bari connettendosi direttamente ad internet (tramite IP masquerading) per scaricare le pagine web:

/sbin/ipchains -A input -s 192.168.1.1/24 -d ! 192.168.1.1 80 -p TCP -j REDIRECT 8080

Se tutto funziona, si aggiungano queste righe alla fine del nostro script di inizializzazione. Di solito è `/etc/rc.d/rc.local`.

Si potrebbe pensare di bloccare il traffico `ftp` (porte 20 e 21) per forzare gli utenti della nostra LAN ad usare Squid, ma questa non è una buona idea per almeno due ragioni:

- Squid è un proxy http con supporto `ftp`, non un proxy `ftp` vero e proprio. Può scaricare da `ftp`, può anche inviare verso alcuni server `ftp`, ma non può eliminare o rinominare file su server `ftp` remoti.

  Bloccando le porte 20 e 21, non si potranno eliminare o rinominare file su server `ftp` remoti.

- IE5.5 ha un bug: non usa il proxy per ottenere la directory `ftp`. Si connette invece direttamente attraverso l'IP masquerading.

  Bloccando le porte 20 e 21 non si potrà navigare nelle directory `ftp` usando IE5.5.

Quindi si dovranno bloccare i download `ftp` voluminosi usando altri metodi. Tratteremo questo argomento nel capitolo 4.

3.3.2. Linux con kernel 2.4.x (iptables)

Ci si assicuri che nessuno bari usando un proxy che non sia il nostro. Proxy pubblici di solito usano le porte 3128 e 8080:

/sbin/iptables -A FORWARD -s 192.168.1.1/24 -d ! 192.168.1.1 --dport 3128 -p TCP -j DROP

/sbin/iptables -A FORWARD -s 192.168.1.1/24 -d ! 192.168.1.1 --dport 8080 -p TCP -j DROP

Si dovrà anche fare in modo che nessuno bari connettendosi direttamente ad internet (tramite IP masquerading) per scaricare le pagine web:

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 8080

Se tutto funziona, si aggiungano queste righe alla fine del nostro script di inizializzazione. Di solito è `/etc/rc.d/rc.local`.

Si potrebbe pensare di bloccare il traffico `ftp` (porte 20 e 21) per forzare gli utenti della nostra LAN ad usare Squid, ma questa non è una buona idea per almeno due ragioni:

- Squid è un proxy http con supporto `ftp`, non un proxy `ftp` vero e proprio. Può scaricare da `ftp`, può anche inviare verso alcuni server `ftp`, ma non può eliminare o rinominare file su server `ftp` remoti.

  Bloccando le porte 20 e 21 non si potranno eliminare o rinominare file su server `ftp` remoti.

- IE5.5 ha un bug: non usa il proxy per ottenere la directory `ftp`. Si connette invece direttamente attraverso l'IP masquerading.

  Bloccando le porte 20 and 21, non si potrà navigare nelle directory `ftp` usando IE5.5.

Si dovranno quindi bloccare i download `ftp` voluminosi usando altri metodi. Tratteremo questo argomento nel capitolo 4.

4. Occuparsi di altri protocolli ad alto traffico usando CBQ

Si deve fare attenzione al fatto che gli utenti della nostra LAN possono rendere vani i nostri sforzi del capitolo 3 se usano Napster, Kazaa o Realaudio. Ci si deve anche ricordare che non si è bloccato il traffico `ftp` nella sezione 3.3.

Si raggiungerà l'obiettivo in un modo diverso: non limitando direttamente il download, ma indirettamente. Se la nostra interfaccia internet è `ppp0` e quella LAN è `eth0`, si dovrà limitare il traffico uscente sull'interfaccia `eth0`, così risulterà limitato anche il traffico entrante in `ppp0`.

Per far ciò si dovrà fare conoscenza con CBQ e con lo script `cbq.init`. Può essere scaricato da ftp://ftp.equinox.gu.net/pub/linux/cbq/. Si dovrà scaricare `cbq.init-v0.6.2` e copiarlo in `/etc/rc.d/`.

Servirà anche `iproute2`. È compreso in tutte le distribuzioni linux.

Si dia ora un'occhiata nella directory `/etc/sysconfig/cbq/`. Ci dovrebbe essere un file di esempio, che dovrebbe funzionare con `cbq.init`. Se non è lì, probabilmente non è incluso nel kernel né è presente come modulo. Bene, in questo caso semplicemente si dovrà creare quella directory, metterci dentro i file d'esempio forniti qui sotto e controllare se in questo modo funziona.

4.1. FTP

Nel capitolo 3 non è stato bloccato FTP per due motivi: per permettere l'upload e per fare in modo che gli utenti con un IE5.5 bacato possano navigare nelle directory `ftp`. In definitiva, i nostri web browser ed i client `ftp` dovrebbero fare i download tramite il nostro proxy Squid, mentre upload/rinominare/eliminare file via `ftp` dovrebbe essere fatto attraverso l'IP masquerading.

Si crei un file chiamato `cbq-10.ftp-network` nella directory `/etc/sysconfig/cbq/`:

 # touch /etc/sysconfig/cbq/cbq-10.ftp-network

Si inseriscano nel file le righe seguenti:

```
DEVICE=eth0,10Mbit,1Mbit
RATE=15Kbit
WEIGHT=1Kbit
PRIO=5
RULE=:20,192.168.1.0/24
RULE=:21,192.168.1.0/24
```

La descrizione di queste righe si trova nel file `cbq.init-v0.6.2`.

Quando si fa partire lo script `/etc/rc.d/cbq.init-v0.6.2`, esso leggerà il file di configurazione posto in `/etc/sysconfig/cbq/`:

 # /etc/rc.d/cbq.init-v0.6.2 start

Se tutto funziona, si aggiunga `/etc/rc.d/cbq.init-v0.6.2 start` alla fine dello script di inizializzazione. Di solito è `/etc/rc.d/rc.local`.

Grazie a questo comando, il nostro server non invierà dati `ftp` attraverso `eth0` a più di 15Kbit/s, di conseguenza non scaricherà dati `ftp` da internet a più di 15Kbit/s. Gli utenti della nostra LAN vedranno che è più efficiente usare il proxy Squid per fare download `ftp`. Riusciranno anche a navigare nelle directory `ftp` usando il loro IE5.5 bacato.

C'è un altro bug in IE5.5: quando si clicca con il bottone destro su un file in una directory `ftp` e poi si seleziona "Copia nella Cartella", il file viene scaricato non attraverso il proxy, ma direttamente attraverso l'IP masquerading, evitando così Squid con i delay pool.

4.2. Napster, Realaudio, Windows Media e altre questioni

Qui l'idea è analoga a quanto fatto per `ftp`; semplicemente si aggiungerà una porta, impostando la velocità voluta.

Si crei un file chiamato `cbq-50.napster-network` nella directory `/etc/sysconfig/cbq/`:

 # touch /etc/sysconfig/cbq/cbq-50.napsterandlive

Si inseriscano queste righe nel file:

```
DEVICE=eth0,10Mbit,1Mbit
RATE=35Kbit
WEIGHT=3Kbit
PRIO=5
# Windows Media Player.
RULE=:1755,192.168.1.0/24
# Real Player usa le porte TCP 554, per l'UDP usa porte diverse,
# ma generalmente RealAudio in UDP non consuma troppa banda.
RULE=:554,192.168.1.0/24
RULE=:7070,192.169.1.0/24
# Napster usa le porte 6699 e 6700, forse anche altre?
RULE=:6699,192.168.1.0/24
RULE=:6700,192.168.1.0/24
# Audiogalaxy usa le porte da 41000 fino a probabilmente 41900,
# ce ne sono molte, quindi si tenga presente che qui non sono elencate tutte.
# Inutile ripetere 900 volte la stessa riga.
# Semplicemente si escludano le porte 410031-41900 usando
# ipchains o iptables.
RULE=:41000,192.168.1.0/24
RULE=:41001,192.168.1.0/24
# ...continua da 41001 a 41030...
RULE=:41030,192.168.1.0/24
# Qualche utente ingegnoso potrebbe connettersi a server SOCKS per Napster,
# Audiogalaxy ecc.; è una buona idea far così anche quando si usa un proxy SOCKS
# locale
RULE=:1080,192.168.1.0/24
# Si aggiungano tutte le porta che servono; si potranno facilmente controllare
# le porte usate con IPTraf
#RULE=:port,192.168.1.0/24
```

Ci si ricordi di bloccare le rimanenenti porte usate da Audiogalaxy (41031-41900), usando ipchains (con i kernel 2.2.x) o iptables (con i kernel 2.4.x).

Kernel 2.2.x.

/sbin/ipchains -A input -s 192.168.1.1/24 -d ! 192.168.1.1 41031:41900 -p TCP -j REJECT

Kernel 2.4.x.

/sbin/iptables -A FORWARD -s 192.168.1.1/24 -d ! 192.168.1.1 --dport 41031:41900 -p TCP -j REJECT

Ci si ricordi di aggiungere una riga appropriata allo script di inizializzazione.

5. FAQ

5.1. È possibile limitare la banda utente per utente con i delay pool?

Sì. Si faccia riferimento al file `squid.conf` originale e si controlli la documentazione di Squid su http://www.squid-cache.org.

5.2. Come posso far funzionare wget con Squid?

È semplice. Si crei un file chiamato `.wgetrc` e lo si metta nella home directory. Si inseriscano nel file le righe seguenti ed il gioco è fatto!

```
HTTP_PROXY=192.168.1.1:8080
FTP_PROXY=192.168.1.1:8080
```

Lo si può far funzionare globalmente per tutti gli utenti; si digiti `man wget` per sapere come fare.

5.3. Ho impostato il mio server SOCKS in ascolto sulla porta 1080 ed ora non riesco più a collegarmi ad alcun server IRC.

Qui ci possono essere due situazioni.

Una è quando il proxy SOCKS è "open relay", il che significa che chiunque può usarlo da qualsiasi parte del mondo. Questo è un problema di sicurezza. Si dovrebbe ricontrollare la configurazione del proxy SOCKS; in genere i server irc non permettono il collegamento di server SOCKS aperti.

In caso si sia sicuri che il proprio server SOCKS non è aperto, si potrebbe comunque essere impossibilitati a collegarsi a qualche server irc: la maggior parte delle volte questo avviene perché essi controllano che sul client che si sta connettendo il server SOCKS non stia girando sulla porta 1080. In questo caso bisognerà semplicemente riconfigurare SOCKS in modo che lavori su una porta diversa. Si dovrà anche riconfigurare il software della LAN in modo che usi server SOCKS e porta appropriati.

5.4. Non voglio che Kazaa o Audiogalaxy occupino tutta la mia banda di ingresso.

Effettivamente questo può essere fastidioso, ma è facile da risolvere.

Si crei un file chiamato, ad esempio, `/etc/sysconfig/cbq/cbq-15.ppp`.

Si inseriscano nel file le righe seguenti e Kazaa o Audiogalaxy scaricheranno a non più di circa 15 kbits/s. Presuppongo che l'interfaccia verso internet sia ppp0.

```
DEVICE=ppp0,115Kbit,11Kbit
RATE=15Kbit
WEIGHT=2Kbit
PRIO=5
TIME=01:00-07:59;110Kbit/11Kbit
RULE=,:21
RULE=,213.25.25.101
RULE=,:1214
RULE=,:41000
```

```
RULE=,:41001
# E così via fino a :41030
RULE=,:41030
```

5.5. Il mio server di posta in uscita consuma tutta la banda.

Si può limitare il traffico SMTP, Postfix, Sendmail, o qualsiasi altro, in un modo simile a quello della risposta precedente. Si deve solo modificare o aggiungere un regola:

```
RULE=,:25
```

Inoltre, se si ha un SMTP server, si possono forzare gli utenti della LAN ad usarlo, anche se questi hanno impostato i loro server SMTP a smtp.qualche.server! Lo si può fare in modo trasparente, come è stato fatto in precedenza con Squid.

5.6. Posso limitare il mio server FTP o WWW in un modo simile a quello mostrato nella domanda precedente?

In generale si può fare, ma di solito questi server hanno la loro proprie configurazioni per la limitazione di banda, quindi sarebbe probabilmente meglio andare a cercare nella loro documentazione.

Kernel 2.2.x

/sbin/ipchains -A input -s 192.168.1.1/24 -d ! 192.168.1.1 25 -p TCP -j REDIRECT 25

Kernel 2.4.x

/sbin/iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 25 -j REDIRECT --to-port 25

Ci si ricordi di aggiungere una riga appropriata allo script di inizializzazione.

5.7. È possibile limitare la banda utente per utente con lo script cbq.init?

Sì. Si guardi nello script: ci sono alcuni esempi.

5.8. Ogni volta che lancio `cbq.init`, dice che non trova `sch_cbq`.

Probabilmente manca il modulo CBQ nel sistema. Se si ha CBQ compilato nel kernel, si commentino le seguenti righe nello script `cbq.init-v0.6.2`.

```
### If you have cbq, tbf and u32 compiled into kernel, comment it out
#for module in sch_cbq sch_tbf sch_sfq sch_prio cls_u32; do
#        if ! modprobe $module; then
#                echo "**CBQ: could not load module $module"
#                exit
#        fi
#done
```

5.9. CBQ a volte non funziona senza alcun motivo.

Generalmente questo non dovrebbe accadere. Talvolta si possono osservare download massivi, anche se si crede di aver bloccato tutte le porte usate da Napster o da Audiogalaxy. Beh, c'è sempre un'altra porta aperta per il download. Per trovarla si può usare IPTraf. Siccome potrebbero esserci centinaia di queste porte, potrebbe rivelarsi un compito davvero arduo. Per renderlo più semplice, si potrebbe considerare l'idea di installare un proxy SOCKS; Napster, Audiogalaxy e molti altri programmi possono usare un proxy SOCKS, così è più semplice avere a che fare con una sola porta, invece di dover gestire migliaia di altre possibilità (la porta SOCKS standard è la 1080, usando un proxy SOCKS locale la si può impostare in modo differente, oppure si possono lanciare diverse istanze del proxy SOCKS su porte diverse). Non si dimentichi di chiudere il traffico su tutte le porte e di lasciare aperte le porte tipo la 25 e la 110 (SMTP e POP3) ed altre che si ritengono utili. Si può trovare un link a Nylon (un proxy SOCKS) alla fine di questo HOWTO.

5.10. I delay pool sono stupidi; perché non posso scaricare qualcosa a piena velocità quando sono l'unico ad usare la rete?

Purtroppo non ci si può far molto.

L'unica cosa che si può fare è usare il comando **cron** e riconfigurare il tutto, ad esempio, all'una di mattina, in modo che Squid non usi i delay pool, quindi riconfigurarlo di nuovo, diciamo alle 7:30, per usare i delay pool.

Per fare questo, si creino due file di configurazione separati, chiamati per esempio `squid.conf-day` e `squid.conf-night`, e li si metta in `/opt/squid/etc/`.

`squid.conf-day` sarà la copia esatta del file di configurazione creato in precedenza.

`squid.conf-night`, al contrario, non avrà alcuna riga di delay pool, così tutto ciò che si deve fare è commentarle.

La cosa successiva da farsi è impostare le voci di `/etc/crontab` correttamente.

Si modifichi `/etc/crontab` inserendo le seguenti righe:

```
#SQUID - cambio di configurazione per notte e giorno
01 9 * * * root /bin/cp -f /opt/squid/etc/squid.conf-day /opt/squid/etc/squid.conf; /opt/squid/bin/squid -k reconfigure
59 23 * * * root /bin/cp -f /opt/squid/etc/squid.conf-night /opt/squid/etc/squid.conf; /opt/squid/bin/squid -k reconfigure
```

5.11. I miei download si bloccano alle 23:59 per colpa di "acl day time 09:00-23:59" in squid.conf. Posso farci qualcosa?

Si può risolvere eliminando quella riga acl da squid.conf e anche le righe "delay_access 2 allow day; delay_access 2 deny !day".

Quindi si provi a farlo con **cron** come visto nella domanda precedente.

5.12. I log di Squid continuano a crescere molto velocemente; posso risolvere in qualche modo?

Effettivamente, più numerosi sono gli utenti e più informazioni (talvolta utili) saranno messe nel log.

Il modo migliore per risolvere alla radice il problema sarebbe quello di usare **logrotate**, ma serve un piccolo trucco per farlo funzionare correttamente con Squid: impostazioni appropriate per **cron** e **logrotate**.

Voci per `/etc/crontab`:

```
#SQUID - logrotate
01 4 * * * root /opt/squid/bin/squid -k rotate; /usr/sbin/logrotate /etc/logrotate.conf; /bin/rm -f /var/log/squid/*.log.0
```

Qui imponiamo a **logrotate** di partire ogni giorno alle 04:01, quindi bisognerà eliminare ogni altre richiesta di esecuzione di **logrotate**, ad esempio in `/etc/cron.daily/`.

Voci per `/etc/logrotate.d/syslog`:

```
#SQUID logrotate - manterra' i log per 40 giorni
/var/log/squid/*.log.0 {
rotate 40
compress
daily
postrotate
/usr/bin/killall -HUP syslogd
endscript
}
```

5.13. CBQ è stupido: perché non posso scaricare qualcosa a piena velocità quando sono l'unico ad usare la rete?

Si può fare, fortunello!

Ci sono due modi per farlo.

Il primo modo è quello semplice, simile alla soluzione che abbiamo usato per Squid. Si inserisca una riga simile a quella qui sotto nei file di configurazione di CBQ in `/etc/sysconfig/cbq/`:

```
TIME=00:00-07:59;110Kbit/11Kbit
```

Si può avere più di un paramentro TIME nei file configurazione di CBQ.

Bisogna comunque fare attenzione, poiché c'è un piccolo bug nello script `cbq.init-v0.6.2`: non è possibile impostare alcuni orari, ad esempio 00:00-08:00! Per essere sicuri che tutto funzioni correttamente si faccia partire lo script `cbq.init-v0.6.2`, quindi, all'interno dell'intervallo che è stato impostato, si digiti

/etc/rc.d/cbq.init-v0.6.2 timecheck

L'output dovrebbe assomigliare a questo:

```
[root@mangoo rc.d]# ./cbq.init start; ./cbq.init timecheck **CBQ: 3:44: class 10 on
eth0 changed rate (20Kbit -> 110Kbit) **CBQ: 3:44: class 40 on ppp0 changed rate
(15Kbit -> 110Kbit) **CBQ: 3:44: class 50 on eth0 changed rate (35Kbit -> 110Kbit)
```

In questo esempio qualcosa è andato storto, probabilmente nel secondo file di configurazione in `/etc/sysconfig/cbq/` (secondo a contare dal numero più piccolo nel nome):

```
[root@mangoo rc.d]# ./cbq.init start; ./cbq.init timecheck **CBQ: 3:54: class 10 on
eth0 changed rate (20Kbit -> 110Kbit) ./cbq.init: 08: value too great for base (error
token is "08")
```

Il secondo modo per rendere CBQ più intelligente è più difficile: non dipende dall'orario. Si possono trovare utili informazioni in "The Linux 2.4 Advanced Routing HOWTO" e giocherellare un po' con il comando `tc`.

6. Varie

6.1. Risorse utili

Squid Web Proxy Cache

http://www.squid-cache.org

Manuale di configurazione di Squid 2.4 Stable 1

http://www.visolve.com/squidman/Configuration%20Guide.html

http://www.visolve.com/squidman/Delaypool%20parameters.htm

FAQ di Squid

http://www.squid-cache.org/Doc/FAQ/FAQ-19.html#ss19.8

Script cbq-init

ftp://ftp.equinox.gu.net/pub/linux/cbq/

Linux 2.4 Advanced Routing HOWTO

http://www.tldp.org/HOWTO/Adv-Routing-HOWTO.html

Controllo del traffico (in polacco)

http://ceti.pl/~kravietz/cbq/

Securing and Optimizing Linux Red Hat Edition - A Hands on Guide

http://www.tldp.org/guides.html

IPTraf

http://cebu.mozcom.com/riker/iptraf/

IPCHAINS

http://www.tldp.org/HOWTO/IPCHAINS-HOWTO.html

Nylon socks proxy server

http://mesh.eecs.umich.edu/projects/nylon/

Traduzione indonesiana di questo HOWTO, di Rahmat Rafiudin *mjl_id@yahoo.com* (mailto:mjl_id@yahoo.com)

http://raf.unisba.ac.id/resources/BandwidthLimitingHOWTO/index.html