Limitar el ancho de banda COMO

Tomasz Chmielewski
Autor

tch@metalab.unc.edu

Jose M. Fernández Navarro
Traductor

elzo@iespana.es

En este documento se describe cómo configurar su servidor bajo Linux para limitar el ancho de banda de las descargas o del tráfico entrante y cómo usar su conexión a Internet de una manera más eficiente.

1. Introducción

El propósito de esta guía es proporcionar una solución sencilla para limitar el tráfico entrante evitando así que los usuarios de nuestra LAN consuman todo el ancho de banda de nuestra conexión a Internet.

Esto resulta útil cuando nuestra conexión es lenta o nuestros usuarios suelen descargar toneladas de mp3s y las iso de las últimas distribuciones de Linux.

1.1. Nuevas versiones de este documento

Siempre puede ver la última versión de este documento en la World Wide Web en la URL http://www.linuxdoc.org.

Las nuevas versiones de este documento también se colocarán en varios sitios web y FTP de Linux, incluyendo la página del LDP en http://www.linuxdoc.org.

1.2. Rechazo de responsabilidad

Ni el autor ni los distribuidores ni cualquier otro contribuyente a este COMO se hacen de manera alguna responsables de los daños físicos, económicos, morales o de cualquier otra índole en el que se haya podido incurrir a partir de las sugerencias de este texto.

1.3. Copyright y licencia

1.4. Retroalimentación y correcciones

Si tiene cualquier pregunta o comentario sobre este documento póngase en contacto con Tomasz Chmielewski en *tch@metalab.unc.edu* (mailto:tch@metalab.unc.edu). Cualquier sugerencia o crítica será bienvenida. Si encuentra algún fallo o errata en este documento (y encontrará muchos porque el inglés no es mi lengua materna) hágamelo saber para que pueda corregirlo en la siguiente versión. Gracias.

1.5. Gracias

Me gustaría darle las gracias a Ami M. Echeverri lula@pollywog.com quien me ayudó a convertir el COMO a SGML y a corregir algunos errores. También quiero darle las gracias a Ryszard Prosowicz prosowicz@poczta.fm por sus útiles sugerencias.

2. Antes de que empecemos

Imaginemos la siguiente situación:

- Disponemos de una conexión a Internet por módem de 115,2 kbits/s ppp (115,2/10 = 11,5 kbytes/s). Nota: con conexiones eht0 (tarjeta de red) dividiríamos 115,2 por 8; con ppp lo dividimos por 10 a causa de los bits de inicio/parada (8 + 1 + 1 = 10).
- Tenemos algunas máquinas en una LAN y sus usuarios llevan a cabo enormes descargas todo el tiempo.
- Queremos que las páginas web se carguen rápido, no importa cuántas descargas se estén llevando a cabo.
- Nuestra interfaz de Internet es **ppp0**.
- Nuestra interfaz de la LAN es **eth0**.
- Nuestra red es 192.168.1.0/24

2.1. Lo que necesitamos

Lo crea o no, modificar el tráfico entrante es una tarea sencilla y no tiene que leer toneladas de libros sobre enrutado o algoritmos de encolado.

Para hacer que funcione necesitamos al menos el proxy Squid; si queremos afinar en su configuración tendremos que familiarizarnos con ipchains o iptables y CBQ.

Para comprobar los resultados podemos usar IPTraf.

2.2. ¿Cómo funciona?

Squid es probablemente el servidor proxy HTTP más avanzado disponible para Linux. Puede ahorrarnos a ahorrar ancho de banda de dos maneras:

- La principal es una característica común a todos los servidores proxy -- guardan las páginas descargadas, las imágenes y otros objetos en memoria o en un disco. Por esto, si dos personas solicitan la misma página no se descargará de Internet sino del proxy local.
- Aparte del cacheado normal, Squid dispone de una prestación especial conocida como [pools: colas?] con retraso. Gracias a las [pools] con retraso es posible limitar el tráfico de internet de una manera razonable dependiendo de las llamadas ‘palabras mágicas’ existentes en cualquier URL. Por ejemplo, una palabra mágica podría ser ’.mp3’, ’.exe’ o ’.avi’, etc. Cualquier parte distintiva de una URL (como .avi) puede definirse como una palabra mágica.

Con eso podemos decirle a Squid que descargue este tipo de archivos a una velocidad específica (en nuestro ejemplo, será a 5 kbytes/s). Si los usuarios de nuestra LAN descargan archivos al mismo tiempo lo harán a 5 kbytes/s entre todos, dejando así ancho de banda suficiente para las páginas web, el correo, las news, el irc, etc.

Evidentemente, Internet no se usa únicamente para descargar archivos a través de páginas web (http o ftp). Más adelante veremos cómo limitar el ancho de banda para Napster, Realaudio y otras posibilidades.

3. Instalar y configurar el software necesario

Aquí explicaremos cómo instalar el software necesario de forma que podamos limitar el ancho de banda y observar su uso.

3.1. Instalar Squid con la prestación delay pools

Como mencioné antes, Squid tiene una prestación que se conoce como delay pools y que nos permite controlar el ancho de banda de bajada. Desgraciadamente, en la mayoría de las distribuciones Squid viene sin esa característica.

Así que si tiene Squid ya instalado siento desanimarle pero tendrá que desinstalarlo e instalarlo de nuevo con las delay pools activadas de la manera que explico más adelante.

1. Para obtener un mejor rendimiento de nuestro proxy Squid lo mejor es crear una partición aparte para su caché y llamarla /cache/. Su tamaño debería ser de alrededor de 300 megabytes dependiendo de sus necesidades.

   Si no sabe cómo crear una partición puede crear el directorio /cache/ en su partición raíz pero el rendimiento de Squid podrá disminuir notablemente.

2. Añadimos un usuario 'squid' seguro:

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

   Nadie puede entrar en el sistema como squid, ni siquiera root.

3. Descargamos las fuentes de Squid de http://www.squid-cache.org

   Cuando estaba escribiendo este COMO la última versión era Squid 2.4 estable 1:

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

4. Desempaquetamos todo en `/var/tmp`:
5. `# tar xzpf squid-2.4.STABLE1-src.tar.gz`
6. Compilamos e instalamos Squid (todo va en una sola línea):

   ```
   # ./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. Configurar Squid para poder usar la prestación de las delay pools

1. Configuramos nuestro archivo squid.conf (localizado en /opt/squid/etc/squid.conf):

```
#squid.conf
#Todas las opciones de este archivo se encuentran muy bien documentadas en el
#propio squid.conf así
#como en http://www.visolve.com/squidman/Configuration%20Guide.html

#
#Los puertos por los que escuchará nuestro Squid.
http_port 8080
icp_port 3130
#los cgi-bin no se cachearán.
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
#La memoria que usará Squid. Bueno, Squid usará mucha más que ésa.
cache_mem 16 MB
#250 significa que Squid usará 250 megabytes de espacio en disco.
cache_dir ufs /cache 250 16 256

#Lugares en los que irán los archivos de bitácora de Squid.
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
#Cuántas veces rotar los archivos de bitácora antes de borrarlos.
#Acuda a la FAQ para más información.
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

#Configure esto si quiere que su proxy funcione de manera transparente.
#Eso significa que por lo general no tendrá que configurar todos los
#navegadores de sus clientes, aunque tiene algunos inconvenientes.
#Si deja esto sin comentar no pasará nada peligroso.
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
```

```
httpd_accel_uses_host_header on

#Todos los usuarios de nuestra LAN serán vistos por los servidores web
#externos como si usasen Mozilla en Linux. :)
anonymize_headers deny User-Agent
fake_user_agent Mozilla/5.0 (X11; U; Linux i686; en-US; rv:0.9.6+) Gecko/20011122

#Para acelerar aún más nuestra conexión ponemos dos líneas similares a las
#de más abajo. Apuntarán a un servidor proxy [parent] que usará nuestro propio
#Squid. No olvide cambiar el servidor por uno más rápido para usted.
#Puede utilizar ping, traceroute y demás herramientas para comprobar la
#velocidad. Asegúrese de que los puerto http e icp son los correctos.

#Descomente las líneas que comienzan por "cache_peer" de ser necesario.
#Éste es el proxy que va a usar para todas las conexiones...
#cache_peer w3cache.icm.edu.pl parent 8080 3130 no-digest default

#...excepto para las direcciones e IPs que comiencen por "!".
#No es buena idea usar un mayor
#cache_peer_domain w3cache.icm.edu.pl !.pl !7thguard.net !192.168.1.1

#Esto resulta útil cuando queremos usar el Cache Manager.
#Copie cachemgr.cgi al cgi-bin de su servidor web.
#Podrá acceder a él una vez lo haya hecho introduciendo en un navegador
#la dirección http://su-servidor-web/cgi-bin/cachemgr.cgi
cache_mgr your@email
cachemgr_passwd secret_password all

#Éste es el nombre de usuario con el que trabajará nuestro Squid.
cache_effective_user squid
cache_effective_group squid

log_icp_queries off
buffered_logs on


#####DELAY POOLS
#Ésta es la parte más importante para configurar el tráfico entrante con
#Squid. Para una descripción detallada acuda al archivo squid.conf o a la
#documentación de http://www.squid-cache.org

#No queremos limitar las descargas en nuestra red local.
acl magic_words1 url_regex -i 192.168

#Queremos limitar la descarga de este tipo de archivos
#Ponga todo esto en una única línea
acl magic_words2 url_regex -i ftp .exe .mp3 .vqf .tar.gz .gz .rpm .zip .rar .avi .mpeg .
.ram .rm .iso .raw .wav .mov
#No bloqueamos .html, .gif, .jpg y archivos similares porque por lo general
#no consumen demasiado ancho de banda.


#Queremos limitar el ancho de banda durante el día permitiendo
```

```
#el ancho de banda completo durante la noche.
#¡Cuidado! con el acl de abajo sus descargas se interrumpirán
#a las 23:59. Lea la FAQ si quiere envitarlo.
acl day time 09:00-23:59

#Tenemos dos delay_pools diferentes
#Acuda a la documentación de Squid para familiarizarse
#con delay_pools y delay_class.
delay_pools 2

#Primer delay pool
#No queremos retrasar nuestro tráfico local
#Hay tres cases de pools; aquí sólo hablaremos de la segunda.
#Primera clase de retraso (1) de segundo tipo (2).
delay_class 1 2

#-1/-1 significa que no hay límites.
delay_parameters 1 -1/-1 -1/-1

#magic_words1: 192.168 que ya hemos puesto antes
delay_access 1 allow magic_words1


#Segundo delay pool.
#Queremos retrasar la descarga de los archivos mencionados en magic_words2.
#Segunda clase de retraso (2) de segundo tipo (2).
delay_class 2 2

#Los números siguientes son valores en bytes;
#Debemos recordar que Squid no tiene en cuenta los bits de inicio/parada
#5000/150000 son valores para la red al completo
#5000/120000 son valores para la IP independiente
#una vez los archivos descargados exceden los 150000 bytes,
#(o el doble o el triple)
#las descargas proseguirán a 5000 bytes/s

delay_parameters 2 5000/150000 5000/120000
#Ya hemos configurado antes el día de 09:00 a 23:59.
delay_access 2 allow day
delay_access 2 deny !day
delay_access 2 allow magic_words2


#EOF
```

Cuando lo hayamos configurado todo debemos asegurarnos de que todo lo que se encuentre en los directorios `/opt/squid` y `/cache` pertenece al usuario '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/

Ahora ya está todo listo para ejecutar Squid. Cuando lo hagamos por primera vez tendremos que crear sus directorios de caché:

 # /opt/squid/bin/squid -z

Ejecutamos Squid y comprobamos si todo funciona correctamente. Una buena herramienta para hacer eso es IPTraf; puede encontrarla en http://freshmeat.net. Asegúrese de haber configurado el proxy adecuado en sus navegadores (192.168.1.1, puerto 8080 en nuestro ejemplo):

 # /opt/squid/bin/squid

Si todo funciona añadimos la línea `/opt/squid/bin/squid` al final de nuestros scripts de inicio. Normalmente puede ser `/etc/rc.d/rc.local`.

Otras opciones útiles de Squid son:

 # /opt/squid/bin/squid -k reconfigure (reconfigures Squid si hicimos cualquier cambio en el archivo squid.conf)

 # /opt/squid/bin/squid -help :) no hace falta explicar qué hace

También puede copiar `cachemgr.cgi` al directorio cgi-bin de su servidor web para utilizar un útil gestor de caché.

3.3. Resolver problemas pendientes

Bien, ya hemos instalado Squid y lo hemos configurado para que use los los delay pools ésos. Apuesto a que nadie quiere restricciones, especialmente los astutos usuarios de nuestra LAN. Seguramente intentarán esquivar nuestras limitaciones para poder descargar sus mp3s favoritos un poco más rápido (y causándonos unos cuantos dolores de cabeza).

Asumo que utiliza enmascarado de IP en su LAN para que sus usuarios puede acceder al IRC, ICQ, correo, etc. Esto está bien, pero debemos asegurarnos de que los usuarios de nuestra LAN utilizarán la caché de nuestro Squid para acceder a la páginas web y el `ftp`.

Podemos resolver la mayoría de estos problemas por medio de `ipchains` (Núcleos Linux 2.2.x) o `iptables` (Núcleos Linux 2.4.x).

3.3.1. Núcleos Linux 2.2.x (ipchains)

Debemos asegurarnos de que nadie hará trampas intentando usar un proxy diferente al nuestro. Los proxies públicos suelen correr en los puertos 3128 y 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

También debemos asegurarnos de que intenta engañarnos conectándose a Internet directamente (Enmascarado de IP) para descargar las páginas web:

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

Si todo funciona añadimos estas líneas al final de nuestros guiones de inicio. Normalmente suele ser `/etc/rc.d/rc.local`.

Podemos pensar en bloquear el tráfico por `ftp` (puertos 20 y 21) para forzar a los usuarios de nuestras LAN a que usen Squid, pero no es una buena idea al menos por dos razones:

- Squid es un proxy http con soporte para `ftp`, no un auténtico proxy `ftp`. Puede descargar desde un `ftp`, también puede "subir" archivos a algún `ftp`, pero no puede borrar/cambiar el nombre de los archivos en servidores `ftp` remotos.

  Cuando bloqueemos los puertos 20 y 21 no podremos borrar/cambiar los nombres de los archivos de servidores `ftp` remotos.

- IE5.5 tiene un error -- no usa un proxy para descargar los directorios `ftp`. En vez de eso se conecta directamente por enmascarado de IP.

  Cuando bloqueemos los puertos 20 y 21 no podremos movernos por directorios `ftp` usando IE5.5.

Así que bloquearemos las descargas excesivas por `ftp` utilizando otros métodos. Nos pondremos con ello en el capítulo 4.

3.3.2. Núcleos Linux 2.4.x (iptables)

Debemos asegurarnos de que nadie intentará hacernos trampa utilizando un servidor proxy distinto al nuestro. Los proxies suelen correr en los puertos 3128 y 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

También debemos asegurarnos de que intenta engañarnos conectándose a Internet directamente (Enmascarado de IP) para descargar las páginas web:

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

Si todo funciona añadimos estas líneas al final de nuestros guiones de inicio. Normalmente suele tratarse de `/etc/rc.d/rc.local`.

Podemos pensar en bloquear el tráfico `ftp` (puertos 20 y 21) para forzar a los usuarios de nuestra LAN a que usen Squid, pero no es una buena idea al menos por dos razones:

- Squid es un proxy http con soporte para `ftp`, no un auténtico proxy `ftp`. Puede descargar archivos desde servidores `ftp`, también puede "subir" archivos a algunos servidores `ftp` pero no puede borrar/cambiar los nombres de los archivos en servidores `ftp` remotos.

  Si bloqueamos los puertos 20 y 21 no podremos borrar/cambiar los nombres de los archivos en servidores `ftp` remotos.

- IE5.5 tiene un error - no usa ningún proxy para descargar los directorios `ftp`. En vez de eso, se conecta directamente a por enmascarado de IP.

  Si bloqueamos los puertos 20 y 21 los usuarios de nuestra LAN no podrán moverse por los directorios `ftp` usando IE5.5.

Así que bloquearemos las descargas excesivas por `ftp` utilizando otros métodos. Nos pondremos con ello en el capítulo 4.

4. Gestionar mediante CBQ otros protocolos que consuman demasiado ancho de banda

Debemos recordar que los usuarios de nuestra LAN pueden echar por tierra todos nuestros esfuerzos en el capítulo 3 si usan Napster, Kazaa o Realaudio. También debemos recordar que en la sección 3.3 no llegamos a bloquear el tráfico por `ftp`.

Lograremos esto de una manera diferente -- no limitando directamente las descargas sino de una manera indirecta. Si nuestro dispositivo de internet es `ppp0` y el de la LAN es `eth0`, limitaremos el tráfico que salga de la interfaz `eth0` limitando así el tráfico entrante por `ppp0`.

Para hacerlo nos familiarizaremos con CBQ y el guión `cbq.init`. Puede obtenerlo de ftp://ftp.equinox.gu.net/pub/linux/cbq/. Descargue `cbq.init-v0.6.2` y colóquelo en `/etc/rc.d/`.

También necesitará instalar `iproute2`. Viene con todas las distribuciones de Linux.

Mire ahora en su directorio `/etc/sysconfig/cbq/`. Debería haber un archivo de ejemplo que debería funcionar con `cbq.init`. Si no está ahí probablemente no lo tiene compilado en su núcleo ni está presente como módulo. Bien, en cualquier caso, simplemente cree ese directorio, coloque en su interior los archivos de ejemplo de más abajo y vea si funcionaría para usted.

4.1. FTP

En el capítulo 3 no bloqueamos el ftp por dos razones: para que pudiésemos subir archivos y para que los usuarios de IE5.5 pudisen moverse por los directorios `ftp`. Con todo, nuestro navegadores y programas de `ftp` deberían poder llevar a cabo las descargas mediante nuestro proxy Squid y subir/renombrar/borrar por `ftp` a través del enmascarado de IP.

Creamos un archivo llamado `cbq-10.ftp-network` en el directorio `/etc/sysconfig/cbq/`:

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

Insertamos en él las siguientes líneas:

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

Encontrará la descripción de estas líneas en el archivo `cbq.init-v0.6.2`

Cuando inicie el guión `/etc/rc.d/cbq.init-v0.6.2` leerá su configuración, que se encuentra en `/etc/sysconfig/cbq/`:

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

Si todo funciona bien añadimos `/etc/rc.d/cbq.init-v0.6.2 start` al final de los guiones de inicio. Normalmente suele ser `/etc/rc.d/rc.local`.

Gracias a esta orden su servidor no enviará datos por `ftp` a través de `eth0` a más de 15kbits/s, por lo que no descargará datos por `ftp` de internet a más de 15kbits/s. Los usuarios de su LAN verán que es más eficiente usar el proxy Squid para realizar las descargas por `ftp`. También podrán moverse por los directorios con su IE5.5.

Hay otro error en IE5.5 - cuando hace clic en un archivo de un `ftp` y elige 'Copiar a una carpeta', el archivo no se descarga a través del proxy sino directamente a través de la IP enmascarada omitiendo así a Squid con sus delay pools.

4.2. Napster, Realaudio, Windows Media y otros asuntos

Aquí la idea es la misma que con `ftp`; simplemente añadimos otro puerto y configuramos una velocidad diferente.

Creamos un archivo llamado `cbq-50.napster-network` en el directorio `/etc/sysconfig/cbq/`:

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

Ponga estas líneas en ese archivo:

```
DEVICE=eth0,10Mbit,1Mbit
RATE=35Kbit
WEIGHT=3Kbit
PRIO=5
#Windows Media Player.
RULE=:1755,192.168.1.0/24
#Real Player usa el puerto TCP 554, para UDP usa puertos diferentes
#pero por lo general RealAudio por UDP no consume demasiado ancho de banda.
RULE=:554,192.168.1.0/24
RULE=:7070,192.169.1.0/24
#Napster usa los puertos 6699 y 6700 ¿quizá algún otro más?
RULE=:6699,192.168.1.0/24
RULE=:6700,192.168.1.0/24
```

```
#Audiogalaxy usa los puertos del 41000 al 41900 más o menos.
#Son muchos así que tenga en cuenta que no los listo todos aquí.
#Repetir cerca de 900 líneas similares no tendría demasiado sentido.
#Simplemente cerraremos los puertos del 410031 al 41900 mediante ipchains o
#iptables.
RULE=:41000,192.168.1.0/24
RULE=:41001,192.168.1.0/24
#seguir del  41001 al 41030
RULE=:41030,192.168.1.0/24
#Algunos usuarios algo astutos pueden conectarse a servidores SOCKS al
#usar Napster, Audiogalaxy, etc. También es una buena idea hacerlo cuando
#ejecute su propio proxy SOCKS.
RULE=:1080,192.168.1.0/24
#Añada cualquier otro puerto que quiera, puede comprobar fácilmente
#cuál usa cada programa con IPTraf.
RULE=:port,192.168.1.0/24
```

No olvide cerrar el resto de puerto (41031-41900) de AudioGalaxy por medio de ipchains (núcleos 2.2.x) o iptables (núcleos 2.4.x).

Núcleos 2.2.x.

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

Núcleos 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

No olvide añadir la línea apropiada a sus guiones de inicio.

5. Preguntas de Uso Frecuente

5.1. ¿Es posible limitar el ancho de banda en base a un usuario mediante las delay pools?

Sí. Mire en el `squid.conf` original y échele un vistazo a la documentación de Squid en http://www.squid-cache.org

5.2. ¿Cómo puedo utilizarwget con Squid?

Es sencillo. Cree un archivo llamado `.wgetrc` y colóquelo en su directorio home. Inserte las siguientes líneas en él ¡y ya está!

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

Puede hacer que funcione globalmente para todos los usuarios, acuda a `man wget` para averiguar cómo.

5.3. Tengo configurado mi propio servidor SOCKS escuchando en el puerto 1080 y ahora ya no puedo conectarme a ningún servidor de irc.

Aquí pueden darse dos casos:

Que su proxy SOCKS esté abierto a reenvíos, eso significa que puede usarlo cualquier desde cualquier parte. Eso es una cuestión de seguridad, por lo que debería volver a comprobar la configuración de su proxy SOCKS - por lo general los servidores de irc no permiten conexiones por parte de servidores SOCKS que permitan el reenvío.

Si está seguro de que su servidor SOCKS no está abierto a reenvíos, quizá aún así no se le permita acceder a ciertos servidores de irc - eso es porque la mayoría de ellos simplemente comprueban si el servidor SOCKS está corriendo en el puerto 1080 de un cliente que esté intentando conectarse. En ese caso simplemente reconfigure el software de su LAN para que use un servidor y un puerto SOCKS adecuados.

5.4. No me gusta que Kazaa ni Audiogalaxy llenen mi ancho de banda de subida.

Puede tratarse de algo doloroso pero existe una solución bien sencilla.

Cree un archivo llamado por ejemplo `/etc/sysconfig/cbq/cbq-15.ppp`.

Inserte en él las siguientes líneas y ni Kazaa ni Audiogalaxy subirán archivos a más de 15 kbits/s. Asumo que su interfaz de salida a internet es 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
#Y así hasta :41030
RULE=,:41030
```

5.5. Mi servidor de correo saliente está consumiendo todo mi ancho de banda

Puede limitar su SMTP Postfix, Sendmail o el que sea de una manera similar a la de la pregunta anterior. Simplemente cambie o añada una regla:

```
RULE=,:25
```

Es más, si tiene un servidor SMTP puede forzar a los usuarios de su LAN local para que lo usen, incluso aunque hayan configurado sus propios servidores SMTP como smtp.algun.servidor Lo haremos de una manera transparente tal y como lo hicimos antes con Squid.

5.6. ¿Puedo limitar mi propio servidor FTP o WEB de una manera similar a como se muestra en la cuestión anterior?

Por lo general sí pero normalmente estos servidores tienen sus propias configuraciones para limitar el ancho de banda por lo que probablemente querrá echar un vistazo a la documentación de los mismos.

Núcleos 2.2.x

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

Núcleos 2.4.x

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

No olvide añadir la línea apropiada a sus guiones de inicio.

5.7. ¿Es posible limitar el ancho de banda en relación con los usuarios mediante el guión cbq.init?

Sí. Observe este guión, hay dos ejemplos.

5.8. Cuando inicio `cbq.init`, dice que falta `sch_cbq`.

Probablemente no tiene CBQ como módulos en su sistema. Si ha compilado CBQ en su núcleo descomente la siguientes líneas en su guión `cbq.init-v0.6.2`.

```
### Si tiene cbq, tbf y u32 compilados en el núcleo descoméntelos aquí
#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 veces no funciona por algún motivo

Normalmente no debería ocurrir. A veces pueden observarse descargas en masa aunque haya creído haber bloqueado todos los puertos que usan Napster o Audiogalaxy. Bien, siempre hay un puerto abierto de más para las descargas en tropel. Para encontrarlo puede usar IPTraf. Como pueden haber posiblemente miles de puertos así resultará una ardua tarea. Para hacerla más sencilla puede considerar ejecutar su propio proxy SOCKS - Napster, Audiogalaxy y muchos otros programas pueden usar proxies SOCKS por lo que resulta mucho más sencillo pelearse con un único puerto que hacerlo con miles de posibilidades (el puerto SOCKS estándar es el 1080, si pone su propio servidor proxy SOCKS podrá configurarlo de manera diferente o hacer correr múltiples instancias escuchando en diferentes puertos). No olvide cerrar todos los puertos al tráfico y dejar abiertos puertos como el 25 o el 110 (SMTP y POP3) y otros que considere podrían resultarle de utilidad. Encontrará un enlace al impotente servidor proxy socks Nylon al final de este COMO.

5.10. Las delay pools son bobas, ¿por qué no me dejan descargar algo utilizando todo el ancho de banda si sólo estoy usando yo la red?

Desgraciadamente no puedes hacer mucho al respecto.

Lo único que puede hacer es usar **cron** y reconfigurarlo a, por ejemplo, la 1:00 AM, de manera que Squid no use delay pools y volver a configurarlo digamos a las 7:30 AM de forma que vuelva a usar las delay pools.

Para hacer esto cree dos archivos de configuración separados llamándoles por ejemplo `squid.conf-day` y `squid.conf-night`, y colóquelos en `/opt/squid/etc/`.

`squid.conf-day` sería una copia exacta de una configuración que hubiésemos creado anteriormente.

`squid.conf-night`, al contrario, no tendría ninguna línea de delay pool, así que lo único que tiene que hacer es descomentarlas.

Lo siguiente que tendría que hacer sería configurar correctamente las entradas de su `/etc/crontab`.

Edite `/etc/crontab` y coloque en él las siguientes líneas:

```
#SQUID - cambio de configuración día/noche
01 9 * * * root /bin/cp -f /opt/squid/etc/squid.conf-day /opt/squid/etc/squid.conf; /opt/sq
59 23 * * * root /bin/cp -f /opt/squid/etc/squid.conf-night /opt/squid/etc/squid.conf; /opt
```

5.11. Mis descargas se interrumpen a las 23:59 con "acl day time 09:00-23:59" en squid.conf. ¿Puedo hacer algo al respecto?

Puede retirar ese acl de su squid.conf así como el "delay_access 2 allow dzien delay_access 2 deny !dzien".

Intente hacerlo ahora con **cron** como en la cuestión anterior.

5.12. Los archivos de bitácora de Squid crecen y crecen muy rápido ¿qué puedo hacer para evitarlo?

De hecho, cuantos más usuarios tenga más información - a veces útil - se registrará.

La mejor manera de arreglarlo sería usar **logrotate**, pero tendrá que emplear un pequeño truco para lograr que funcione con Squid: entradas adecuadas en **cron** y **logrotate**.

Entradas en `/etc/crontab`:

```
#SQUID - logrotate
01 4 * * * root /opt/squid/bin/squid -k rotate; /usr/sbin/logrotate /etc/logrotate.conf; /b
```

Aquí hemos hecho que **logrotate** arranque diariamente a las 04:01 am de forma que elimina cualquier punto de inicio de **logrotate** que pudiese quedar pendiente, por ejemplo de `/etc/cron.daily/`.

Entradas de `/etc/logrotate.d/syslog`:

```
#SQUID logrotate - guardará los archivos de bitácora durante 40 días
/var/log/squid/*.log.0 {
rotate 40
compress
daily
postrotate
/usr/bin/killall -HUP syslogd
endscript
}
```

5.13. CBQ es estúpido; ¿por qué no puedo descargar algo a toda velocidad si sólo yo uso la red?

Está de suerte ¡eso es posible!

Hay dos maneras de conseguirlo.

La primera es la más fácil, similar a la solución del caso Squid. Inserte una línea similar a la de aquí abajo en sus archivos de configuración de CBQ en `/etc/sysconfig/cbq/`:

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

Puede tener múltiples variables TIME en sus archivos de configuración de CBQ.

Eso sí, vaya con ojo porque hay un pequeño fallo en ese `cbq.init-v0.6.2` - no le dejará configurar ciertas horas, como por ejemplo 00:00-08:00 Para asegurarse de que todo funciona correctamente inicie `cbq.init-v0.6.2`, y durante el periodo de tiempo que haya configurado use la orden

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

Esto es un ejemplo de cómo debería ser la salida adecuada:

```
[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)
```

En este ejemplo algo fue mal, probablemente en el segundo archivo de configuración situado en `/etc/sysconfig/cbq/`; segundo contando a partir del número más bajo de su nombre:

```
[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")
```

El segundo método de hacer a CBQ más inteligente es algo más complejo - no está relacionado con el tiempo. Puede leer sobre él en el COMO sobre Enrutado Avanzado en Linux 2.4 y jugar con la orden `tc` command.

6. Miscelánea

6.1. Recursos útiles

Proxy Caché Web Squid

http://www.squid-cache.org

Manual de configuración de Squid 2.4 estable 1

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

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

PUFs sobre Squid

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

Guión cbq-init

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

Enrutado Avanzado en Linux 2.4 COMO

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

Control del tráfico (en polaco)

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

Seguridad y Optimización. Edición para Linux Red Hat - Una Guía Práctica

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

IPTraf

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

IPCHAINS

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

Servidor proxy socks Nylon

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

Traducción indonesia de este COMO por Rahmat Rafiudin *mjl_id@yahoo.com* (mailto:mjl_id@yahoo.com)

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