



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

## Squid y los Delay Pools (64948 lectures)

Per **carlos A. Martinez**, [camarti](#) ()

Creado el 17/03/2006 05:29 modificado el 17/03/2006 05:45

*Los Delays Pools son la alternativa que ofrece Squid para el control del ancho de banda; son una de las herramientas más importantes que existen en Squid. A pesar de ello, están poco documentadas y con frecuencia son mal empleadas. Adicionalmente casi toda la documentación al respecto está en inglés y la información que se encuentra en la Web a veces se contradice o confunde los conceptos. Este es un intento por documentar en nuestro idioma su uso, ventajas e inconvenientes presentando dos ejemplos de implementación en ambientes de producción reales. Este no es un documento técnico y conciso. Muestra su uso exponiendo las situaciones a la cuales se ha enfrentado el autor.*

- [1 Introducción](#)
  - ◆ [1.1 ¿De qué trata esto?](#)
  - ◆ [1.2 Descargo de responsabilidad](#)
  - ◆ [1.3 Audiencia y conocimientos previos](#)
  - ◆ [1.4 Existe un error en este documento o quiero aportar algo ¿dónde lo notifico?](#)
- [2 El problema](#)
- [3 Quiero controlar el canal que usa Squid](#)
  - ◆ [3.1 Clases de Delay Pools](#)
  - ◆ [3.2 Definición de Delay Pools](#)
  - ◆ [3.3 Definición de la clase](#)
  - ◆ [3.4 Parámetros del Delay Pool](#)
  - ◆ [3.5 delay\\_access](#)
  - ◆ [3.6 Nivel del bucket al inicio](#)
  - ◆ [3.7 Monitoreo y prueba](#)
- [4 Implementaciones](#)
  - ◆ [4.1 Primer caso](#)
  - ◆ [4.2 Segundo caso](#)
- [5 Limitaciones](#)
- [6 Conclusiones y pendientes](#)
- [7 Dónde encontrar más información](#)
- [8 Contactos](#)

# 1 Introducción

## 1.1 ¿De qué trata esto?

Si llegó hasta aquí ya sabe que hablaré acerca de los Delays Pools.

Antes de continuar hago una advertencia: La forma en que he escrito este documento, es informal así que no respetaré la manera de escribir los documentos técnicos (escribir siempre en tercera persona, evitar al máximo la familiaridad con el lector, extrema cautela en la escogencia de las palabras, etc.), aunque sí garantizo que lo que aquí aparece lo he tratado de validar de la manera más extensa posible e implementado en entornos de producción. Es decir, pretendo escribir desde la experiencia siempre teniendo en mente los problemas que se presentan cuando se implementa.



Usuarios avanzados de Squid, seguramente no necesitan leer esto. Mejor diviértanse leyendo las tiras de Bilo y Nano o las de Raulito el Friki :-)

## 1.2 Descargo de responsabilidad

Este documento se provee con la esperanza de que sea útil, pero **sin ningún tipo de garantía**. En resumen y para no alargar el asunto, lo que haga con este escrito es completamente su problema. Si los usuarios llegan a tratarlo mal por poner en práctica lo que aquí se expone o las cosas salen mal, es su culpa. Lo único que puedo asegurarle es que este documento (con el estado del arte de la computación y hardware actuales) ocupa espacio en la memoria de su pc (o del pc desde el cual lo está leyendo).

## 1.3 Audiencia y conocimientos previos

Este documento está orientado hacia aquellos usuarios que poseen un conocimiento básico-medio de Squid. No se explicará nada - o muy poco - acerca de las acl's o de cómo compilar Squid. Se asume que tiene conocimientos básicos acerca de las direcciones IP. En la parte de conceptos se trata de abordar el tema de la manera más exhaustiva posible. Esto es intencional y cumple un doble propósito: por una lado busca motivar al lector para que comprenda muy bien cómo funcionan los Delay Pools antes de implementarlos en sus sistemas y por el otro pretende que estos validen, corrijan y amplíen lo que aquí se expone.

## 1.4 Existe un error en este documento o quiero aportar algo ¿dónde lo notifico?

Una de las razones por la cual escribo esto es para que la comunidad de usuarios de Squid me ayude a mejorar las implementaciones que hago en entornos de producción y las compartan con todos. En el mundo de los servidores, el problema no está en instalar sino en configurar. Cualquiera instala un proxy. Las configuraciones pueden funcionar razonablemente bien pero siempre hay espacio para la optimización y el manejo de casos no contemplados o pasados por alto (y ni hablar de los aspectos de seguridad). Pueden escribirme a [camarti\[at\]gmail\[dot\]com](mailto:camarti[at]gmail[dot]com). Tengan la certeza de que sus aportes los implementaré y los daré a conocer a la comunidad.

## 2 El problema

Si está plenamente convencido que los Delay Pools resuelven su problema, salte esta parte y vaya la siguiente sección.

Squid es probablemente el mejor proxy-caché que existe para HTTP. Es bastante estable, fácil de configurar y con algunas precauciones básicas, relativamente seguro. La principal razón por la cual se instala Squid en pequeñas y medianas organizaciones es para disminuir la carga que el tráfico HTTP (y FTP, IM, P2P, etc.) impone sobre el canal de Internet. El caché reduce el uso del canal pero no soluciona el problema. Si un usuario decide bajar la imagen de un DVD o el video porno de 100MB, Este seguramente no estará en caché (a menos que usted haya hecho cosas raras con la política de almacenamiento de objetos grandes). Así, este usuario consume parte del canal disponible durante un largo periodo y afecta el trabajo de los demás. Si el servidor de descarga tiene un buen ancho de banda disponible, este único usuario puede hacer de la experiencia de navegación dentro la organización una verdadera pesadilla. O si un usuario abre 10 ventanas en su navegador simultáneamente puede igualmente disminuir la calidad del servicio e incluso llegar a paralizarlo (y no menciono nada acerca de P2P). Aquí es donde empiezan a aparecer las posibles soluciones que no resuelven nada: aumentar el tamaño máximo de los objetos que se guardan en caché, aumentar el tamaño del caché, comprar más RAM, o cambiar de servidor "porque la pobre máquina con el procesador y disco que tiene no es capaz de manejar el gran volumen de peticiones HTTP recibidas". Si a esto le sumamos un proxy transparente la situación se complica más. ¿Por qué la explicación tan larga? Por las siguientes razones:

1. *Identifique cuál es realmente el problema*, es decir, si un usuario o conjunto reducido de usuarios son los que consumen la mayor parte del canal o es simplemente que de verdad necesita más ancho de banda. Recuerde: GNU/Linux [o el sistema operativo X] y Squid no hacen milagros como la generación espontánea de ancho de banda. Para grupos de 300 usuarios o más se estima un consumo promedio de 2.5Kb/s de bajada por 1.2Kb/s de subida (ojo que son Kilobits no KiloBytes y estas cifras pueden variar de acuerdo al tipo de organización). No espere que un enlace telefónico de 56Kb/s o de banda ancha de 128Kb/s con reuso de 1:4 ó 1:6 haga que 50 usuarios naveguen como si estuvieran pegados directamente al backbone.



2. *Identifique dónde está el cuello de botella*: no todo se le puede achacar "al poco ancho de banda que le compramos al ISP". Puede estar en su LAN o en el propio servidor proxy. con Squid como proxy/caché de Internet usted no controla el tráfico interno.
3. *Revise la instalación de Squid*. Si instala el proxy en modo transparente lo más probable es que usted no haga pasar por él todas las peticiones HTTP y el tráfico de todas las aplicaciones que puedan hablar HTTP. Existen servidores que reciben peticiones en puertos no estándar y a menos que usted viva espiando a los usuarios siempre hay un puerto a la espera de ser redirigido.
4. *Si su problema es controlar el P2P (Kazaa, Emule, Edonkey y demás amigos) o controlar la mensajería instantánea*, no sueñe con detenerlos únicamente con Squid. Usted está buscando en el lugar equivocado: estos engendros los controla con una agresiva política de firewall, parches para el kernel de Linux/iptables (POM de netfilter) y un **proxy NO transparente** con Delay Pools. si esto no es suficiente puede empezar con tc e IMQ en GNU/Linux.
5. *Comprenda muy bien hasta dónde puede llegar hoy con Squid*: nunca instale versiones de desarrollo en entornos de producción o le ponga parches que solo conoce el autor que lo escribió, la madre de este y usted que quiere instalarlo (a menos que sea un versado en el código y sepa realmente qué está cambiando). Nunca pierda de vista la seguridad y la estabilidad. Además, jamás diga a los clientes que su implementación es la solución a todos sus problemas; venda la solución, pero en el camino no arruine su reputación. Primero infórmese acerca de las costumbres de navegación de los usuarios y la cultura organizacional.

## 3 Quiero controlar el canal que usa Squid

Si está plenamente convencido de ello entonces, los Delay Pools son de su interés. Los Delay Pools son la herramienta para llevar a cabo el control de ancho de banda del Proxy (*rate limiting y traffic shaping*). La belleza de estos radica en que controlan el ancho de banda, sin causar penalidades sobre los objetos traídos desde el caché. En lenguaje técnico de Proxy, los Delay Pools afectan los *cache misses*, no los *cache hits*.

Existen tres clases de Delay Pool lo que nos permite tener cierta flexibilidad en su uso. Antes de hablar sobre ellas imagine que un Delay Pool, especialmente uno de la clase 1, es como un tanque de agua el cual tiene un tubo de entrada y otro de salida. El tubo de entrada debido a su diámetro y a la apertura de la llave de paso solo permite que el agua entre a una rata fija. El tubo de salida debido a su gran diámetro no tiene estas limitaciones y puede vaciar el tanque inmediatamente si la llave de paso se abre lo suficiente. Finalmente, el tanque siempre almacena una cantidad máxima de agua (soy perezoso así que haga usted el dibujo en cuestión e imagine que usa el agua que sale por el tubo de salida de todas las formas posibles y visualice lo que pasa dentro del tanque). Una vez hecho esto asocie:

1. El diámetro del tubo de entrada representa el ancho de banda disponible en total. Usted puede abrir la llave tanto como pueda pero la cantidad máxima de agua que sale no sobrepasa un tope máximo.
2. La apertura de la llave de paso de entrada representa el canal destinado para uso del Proxy/Delay Pool respectivo. Esta es la *rata* [de llenado] del Delay Pool.
3. La cantidad máxima de agua que almacena el tanque es el tamaño (*size*) del Delay Pool. Esta es la parte menos comprendida: el size del Delay Pool permite ráfagas (burst) de descarga mientras el Delay Pool se vacía. Cuando el Delay Pool queda con un tamaño de cero solo es posible descargar a la rata definida (de nuevo imagine un caso de uso intensivo del agua del tanque y después de un tiempo piense qué sucede).
4. La apertura de la llave de salida representa su demanda de canal. Entre más abre la llave, más agua intenta obtener. Entre más archivos trate de traer desde Internet, más canal consume.

Ya con esto usted está en total capacidad de usar Delay Pools clase 1.

### 3.1 Clases de Delay Pools

En Squid 2.x existen tres clases de Delay Pool. En la serie 3 (no estable y en desarrollo) existen cinco. nunca he trabajado con Squid 3.x así que no puedo hablar acerca de estos en Squid 3.x. Esta documentación asume que usa Squid 2.x.

#### 3.1.1 Clase 1

El Delay Pool clase 1 define una única estructura de control (en nuestra abstracción un solo tanque). Este limita el uso del canal de manera global sin importar cómo lo usan los clientes internamente o cómo esta definida lógicamente la



LAN. En el inglés técnico se habla de la definición de un único *aggregate bucket*. Esta es la opción indicada si usted desea limitar el ancho de banda que usa Squid, sin importar cómo lo emplean los usuarios.

### 3.1.2 Clase 2

Este es un Delay clase 1 con un 256 Delay Pools clase 1 subordinados a este. En inglés técnico un *aggregate bucket* y 256 *individual buckets*. En nuestra abstracción un tanque principal y 256 tanques secundarios alimentados por el tanque principal. Con este Delay es posible controlar el canal que usan 256 clientes.

¿Cómo se asigna el canal a cada cliente? Squid asume que su LAN es una LAN clase C y usa los últimos 8 bits del número IP del cliente para identificarlo y manejarlo en su *individual bucket* correspondiente. En la práctica solo se pueden controlar 253 clientes descontando la dirección de red, la dirección de broadcast y la dirección del proxy. Note que aquí se empiezan a enredar las cosas: es muy diferente hablar de un cliente, un host y un usuario. Squid puede tener 230 clientes en una red de 600 hosts (equipos) y unos 700 usuarios. Jamás confunda estos conceptos aunque pueden ser equivalentes dependiendo de la situación.

### 3.1.3 Clase 3

Este es un Delay Pool clase 1 con 256 Delay Pools clase 2 subordinados a este. En inglés técnico un *aggregate bucket*, 256 *network buckets*, y 65,536 *individual buckets*. Está orientado para manejar la asignación de ancho de banda en redes clase B. los bits 17 a 24 del número IP identifican la red y los bits 17 a 32 el cliente.

## 3.2 Definición de Delay Pools

Cuando se compila Squid con la opción `-enable-delay-pools` se tiene acceso a esta característica. En `squid.conf` la cantidad de Delay Pools a emplear se define con la directiva `delay_pools`.

Sintaxis:

```
delay_pools N
```

donde  $N > 0$  representa la cantidad de Delay Pools a usar.

Ejemplo:

```
delay_pools 3
```

Con esto se le dice a Squid que se van a usar y definir tres Delay Pools.

## 3.3 Definición de la clase

La clase del Delay Pool se especifica con la directiva `delay_class`.

Syntaxis:

```
delay_class id class
```

donde  $id > 0$ ,  $class = [1|2|3]$ ;  $id$  es el identificador y  $class$  la clase

Ejemplo:

```
delay_class 1 3 ## el Delay Pool número 1 será clase 3
```

```
delay_class 2 1 ## el Delay Pool número 2 será clase 1
```

```
delay_class 3 2 ## el Delay Pool número 3 será clase 2
```



Los Delay Pools no tienen nombre, se identifican con un número que empieza en 1 y termina en N.

### 3.4 Parámetros del Delay Pool

Los parámetros de cada Delay Pool se definen por medio de la directiva `delay_parameters`:

Sintaxis:

```
delay_parameters id rate/size [ rate/size [rate/size]]
```

la valores de *rate* y *size* son dados en Bytes. Por ende no olvide hacer la conversión respectiva de Kbits como le venden el canal a Bytes. Size es dos o tres veces el valor de rate.

Ejemplos:

```
delay_parameters 1 76800/230400 42800/100000 10000/70000
```

Un Delay clase 3 con 600Kb/s (76800B/s) en total para navegación, con un tamaño para ráfagas (burst) globales de 1800Kb (230400Bytes). Para cada subred se asigna un canal máximo de 334.3Kb/s un tamaño para ráfagas de 781.2Kb (100000Bytes) con un ancho de banda para cada host de 78.1Kb/s (10000B/s) con la posibilidad de ráfagas de descargas de 546.8Kb (70000Bytes). Note que los valores para cada subred y host exceden los límites de canal disponible si hay más de 4 clientes navegando en una subred o dos subredes demandan todo el canal asignado. En este caso se produce una condición de competencia y el primero que solicita el canal es el que lo obtiene. Es de suponer que esta asignación es para una organización con usuarios que navegan poco y requieren un buen desempeño al momento de solicitar un archivo. Este es un buen ejemplo de cómo se puede jugar con los parámetros del Delay Pool teniendo en cuenta las costumbres de navegación de la organización.

```
delay_parameters 1 76800/230400
```

Un Delay clase 1. Se usan máximo 600Kb/s de ancho de banda con ráfagas de descarga de 1800Kb. Solo se limita el ancho de banda que usa en total sin importar cómo se distribuye el canal entre los clientes, lo cual da la posibilidad a condiciones de competencia por el ancho de banda en todo momento.

```
delay_parameters 1 340787/1022361 10000/200000
```

Un Delay clase 2. Define que se usarán máximo 2.6Mb/s para navegación con ráfagas de 7.8Mb para una asignación de canal a máximo 256 clientes de 78.1Kb/s con ráfagas de descarga de 195.3Kb (esto es, pueden descargar 195.3Kb a todo lo que de el canal si tienen el *individual bucket* lleno). Este montaje es para una organización cuyos usuarios demandan una gran cantidad de canal. Aquí el peor caso se presenta cuando hay 34 clientes demandando todo el canal asignado de forma continua.

En los Delay Pools clase 2 y clase 3 es posible deshabilitar los *buckets* que no se desea utilizar colocando -1/-1 en el *bucket* correspondiente. Por ejemplo:

*# asigno el canal a hosts en una red clase C sin límite global*

```
delay_parameters 1 -1/-1 10000/200000
```

*# asigno canal en una red clase B a hosts individuales sin límites por subred*

```
delay_parameters 2 340787/1022361 -1/-1 10000/200000
```

*# asigno canal en una red clase B a cada su red sin importar los hosts*

```
delay_parameters 3 340787/1022361 10000/200000 -1/-1
```



## 3.5 delay\_access

Definen por medio de acl's cuáles peticiones pasan por el Delay y cuáles no. Ver los ejemplos de uso en la sección de implementaciones.

Sintaxis:

```
delay_access id allow acl name | deny acl name
```

## 3.6 Nivel del bucket al inicio

La directiva `delay_initial_bucket_level`, controla qué tan lleno estará el bucket al arranque del Squid o cuando se reconfigura. Esto afecta tanto a los *individual buckets* como a los *aggregate buckets*. Note que los *buckets* no se crean sino hasta que son referenciados por primera vez. El valor es un porcentaje.

Sintaxis:

```
delay_initial_bucket_level porcentaje
```

*porcentaje* toma valores entre 0-100

Ejemplo:

```
delay_initial_bucket_level 90 #el bucket inicia un 90% lleno (con un 90% de su size definido)
```

## 3.7 Monitoreo y prueba

Puede monitorear los Delay Pools con el comando:

```
squidclient mgr:delay | less
```

Para probar el funcionamiento de los Delay Pools pruebe a descargar un archivo grande. Una vez haya descargado dos o tres veces el tamaño del *size* del *bucket* respectivo, notará que el archivo baja a la rata fijada. Si puede usar `wget` o `iptraf` para monitorear el ancho de banda usado en la descarga, hágalo. IE y Firefox indican promedios en la velocidad de descarga, y por lo tanto pueden producir la sensación de que los Delay Pools no funcionan bien.

# 4 Implementaciones

## 4.1 Primer caso

Los valores son para una organización con una red clase B con cerca de 1000 usuarios de los cuales solo navegan en Internet unos 300 (esto se logra con autenticación en el Proxy). Los 300 usuarios hacen pocas peticiones al caché. No importa qué descarguen, todo pasa por el mismo Delay Pool.

```
#No pasan por el delay los servidores en la LAN ni los de la DMZ
```

```
#Para eso son las acl's LocalServers y LocalDomain
```

```
delay_class 1 3
```

```
#600Kbits
```

```
delay_parameters 1 76800/240400 42800/100000 10000/70000
```

```
delay_initial_bucket_level 90
```



```
delay_access 1 allow !LocalServers !LocalDomain all
```

```
delay_access 1 deny all
```

## 4.2 Segundo caso

Una organización con una red clase C con usuarios que son grandes consumidores de canal. Se hacen distinciones entre los tipos de archivo que se traen desde Internet. Por un Delay pasan unos tipos de archivo; por otro pasa la navegación en general y por otro pasa un "cliente especial". Como mínimo para esta organización se requieren 2.6Mb/s. Note que con 2.6Mb/s hay bastante reuso (si sumamos todos los anchos de banda asignados a cada Delay obtendríamos 5.4Mb/s). Sin embargo con 2.6Mb/s la gran mayoría de los usuarios (más de 600), están contentos y el canal se usa totalmente en horarios laborales.

```
acl restringir url_regex -i "/etc/squid/extensions"
```

```
acl ClienteEspecial src 192.168.2.12/32
```

```
# cantidad de delay pools
```

```
delay_pools 3
```

```
delay_class 1 2 # delay para extensiones restringidas
```

```
delay_class 2 2 # delay para navegacion en general
```

```
delay_class 3 1 # delay para un cliente especial
```

```
delay_initial_bucket_level 90
```

```
# 2.6Mbits en esencia solo me importa aquí el individual bucket.
```

```
delay_parameters 1 340787/1022361 10000/200000
```

```
# le doy un enlace de 128Kb = 16KB = 16384B a cada IP
```

```
delay_parameters 2 340787/1022361 16384/200000
```

```
# delay para el cliente especial
```

```
delay_parameters 3 26384/300000
```

```
delay_access 3 allow ClienteEspecial !restringir
```

```
delay_access 3 deny all
```

```
delay_access 2 allow all !ClienteEspecial !restringir
```

```
delay_access 2 deny all
```

```
delay_access 1 allow restringir
```

```
delay_access 1 deny all
```

El archivo /etc/squid/extensions contiene una extensión por línea:

```
\.ace$
```

```
\.af$
```



\.afx\$

\.arj\$

\.asf\$

\.asx\$

\.au\$

[...]

\.xla\$

\.xls\$

\.xlt\$

\.xlw\$

\.z\$

\.zip\$

## 5 Limitaciones

No todo son maravillas con los Delay Pools. Estos poseen limitaciones que deben ser tomadas en consideración:

- No comparten ancho de banda. Esta es la principal limitación. Si existe un solo cliente demandando ancho banda de Internet la cantidad máxima de canal que recibe esta limitado por el Delay más restrictivo que lo cobija sin importar que él sea el único haciendo peticiones al caché. Si el cliente tiene asignados 20KB/s y hay 200KB/s sin usar, seguirá recibiendo los *cache misses* a máximo 20KB/s. Si su objetivo es optimizar el uso del canal, por este lado, no hay mucho por hacer.
- No hay garantía de asignación equitativa de canal entre los clientes de un mismo *bucket*. Esto es especialmente relevante en los *aggregate buckets*.
- Están fuertemente orientados hacia el subnetting de la red. Se identifican los clientes por los últimos bits del número IP. Si sus clientes se identifican en Squid por un nombre de usuario y desea asignar canal por usuario y no por IP, mala cosa. Esto es bastante preocupante en el caso de usuarios móviles (principalmente si los usuarios móviles en cuestión son los que firman el cheque). Squid 3.X maneja Delay Pools que trabajan con los usuarios autenticados en Squid.
- Si desea manejar unos valores de *rate/size* en horarios laborales y otros valores en horarios no laborales, en el momento de hacer los cambios puede provocar la interrupción de las descargas. El cambio de valores lo puede realizar con acl's de tiempo o con un reconfigure de Squid a la hora deseada haciéndolo leer un archivo de configuración diferente (dos o tres líneas de cron, dos archivos de configuración de Squid que solo cambian en los valores *rate/size* del Delay, un enlace simbólico y un script de bash hacen el trabajo).

## 6 Conclusiones y pendientes

A pesar de ser un documento muy largo para un tema tan específico, todavía hay para hacer. Por ejemplo, mejorar las acl's que manejan las extensiones/tipos de archivo (si cambia la extensión se puede evitar el Delay correspondiente), verificar que las acl's hagan realmente lo que se pretende, hablar de los Delay Pools en Squid 3.x y si estos funcionan con IPv6 (lo cual dudo); hablar acerca de valores estadísticos acerca del tipo de organización y el consumo por usuario con miras al dimensionamiento de los enlaces de Internet y la asignación de ancho de banda, etc.

Respecto a conclusiones y notas finales podemos decir:





- Son un método para mantener a raya a los usuarios problemáticos más que para optimizar el uso del canal. En este caso me refiero a los Delay Pools como un método para repartir la pobreza :-). Los Delay Pools se implementan cuando el ancho de banda es un recurso escaso, costoso y hay usuarios abusivos.
- Trabajan empleando lo que técnicamente se denomina reuso del canal (sí, como la banda ancha). Confía en que todos los usuarios/clientes no hacen peticiones al Proxy simultáneamente. Por eso se habla también de los 2.5Kb/1.2Kb downstream/upstream por usuario.
- Pueden limitar el canal que consume el P2P al limitar el ancho de banda asignado a un cliente, pero NO eliminan el tráfico P2P. No espere tampoco que cerrar los puertos de P2P que se encuentran en Internet, elimina el problema. Algunos pueden encapsular el tráfico por HTTP y pasar a través del Proxy.
- Si desea optimizar el uso del canal y evitar abusos, debe considerar una combinación de Proxy no transparente, Delay Pools, tc (iproute) e IMQ.

## 7 Dónde encontrar más información

En Internet se encuentra abundante información acerca de Squid, mucha de la cual es de dudosa reputación como este documento. Siempre confíe más en la documentación oficial que en los otros escritos.

- *Squid: The Definitive Guide* By Duane Wessels. Publisher : O'Reilly. ISBN : 0-596-00162-2. Muy bueno. Compre el libro, no sea pirata :-)
- *Web Caching* By Duane Wessels. Publisher: O'Reilly. ISBN: 1-56592-536-X. Algo antiguo pero muy útil en cuanto a la teoría
- <http://squid.visolve.com/squid/index.htm>. Aqué están los manuales de configuración de Squid y papers relativos al tema
- *Enrutamiento avanzado y control de tráfico en Linux / Linux Advanced Routing & Traffic Control HOWTO*. En cualquier idioma, la referencia obligada para aquellos que desean incursionar en el mundo del control del ancho de banda en GNU/Linux

## 8 Contactos

El email al cual pueden escribirme es [camarti\[at\]gmail\[dot\]com](mailto:camarti[at]gmail[dot]com)

---

E-mail del autor: [camarti\\_ARROBA\\_gmail.com](mailto:camarti_ARROBA_gmail.com)

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=2284>