

# **CONFIGURACIÓN DE PARÁMETROS DE DELAY POOLS PARA USO EFICIENTE DEL CANAL DE NAVEGACIÓN. CASO UNIVERSIDAD DE MATANZAS**

**Ing. Darien Menéndez Molina<sup>1</sup>**  
**MSc. Pedro Aniel Sánchez Mirabal<sup>2</sup>**

- 1. Dirección de Redes. Universidad de Matanzas “Camilo Cienfuegos”.*
- 2. Departamento de Informática. Universidad de Matanzas “Camilo Cienfuegos”*

## **Resumen.**

No siempre las instituciones disponen de ancho de banda suficiente para suplir las necesidades de navegación en Internet de los trabajadores en sus labores cotidianas, ya sea por problemas de presupuesto o tecnológicos. Mecanismos como las *delay pools* en las aplicaciones *proxies*, ayudan a controlar el ancho de banda y repartirlo equitativamente por usuarios, pero la selección de los parámetros para su implantación debe realizarse con sumo cuidado, de lo contrario podrían causar un uso ineficiente del canal y por consiguiente brindar un servicio de navegación de mala calidad. En el artículo se muestra cómo se realizó la selección de parámetros adecuados aumentando la eficiencia del empleo del canal.

***Palabras claves:*** *Parámetros; Delay Pools; Configuración; Navegación Internet; Proxy*

---

## Introducción

En el ámbito empresarial e institucional cubano, desde hace algunos años, se viene incrementando el uso de Internet como complemento en las tareas laborales, por lo que son cada vez mayores las instituciones que se conectan a la red de redes. La gran cantidad de información y servicios disponible en las redes de datos representa un beneficio no despreciable para cualquier institución.

En ocasiones por problemas de presupuesto o tecnológicos, resulta difícil obtener un ancho de banda lo suficientemente bueno como para suplir todas las necesidades, por lo que su correcto aprovechamiento es de vital importancia. Un mal aprovechamiento de esos recursos puede conllevar a brindar un servicio de mala calidad y por consiguiente la insatisfacción de todos los usuarios de la red.

### **Caso de Estudio: Universidad de Matanzas “Camilo Cienfuegos”.**

La Universidad de Matanzas “Camilo Cienfuegos” (UMCC) posee para el desempeño de sus funciones un ancho de banda de bajada en Internet de 384 Kbit/s, asignado por el Ministerio de Educación Superior (MES). De todo el canal, 128 Kbit/s se emplean para la recepción de correo del exterior y 256 Kbit/s para la navegación web. Qué estrategia emplear, para lograr un uso eficiente de tales recursos supone un reto, teniendo en cuenta que a la universidad afluyen decenas de profesores e investigadores diariamente, y las necesidades de navegación son muy por encima de las que se puede ofrecer con dicho ancho de banda.

Para lograr tener un control de todas las solicitudes de navegación web que se realizan por parte de los usuarios de la institución, se tiene instalado un servidor proxy, específicamente el Squid-Caché versión 3.1.16. Este servidor es capaz de controlar todas las peticiones realizadas y asignarlas a sus respectivos usuarios del dominio.

Normalmente el acceso a Internet a través de este servidor se realiza por contienda. Los usuarios pelean por obtener el ancho del canal necesario en sus peticiones y el servidor por suplirlas a toda costa, pero como los recursos no son suficientes para suplir las necesidades de navegación, una parte de los usuarios no logra ser atendida. Eso fomenta el uso de aceleradores de descargas por parte de los usuarios, y por consiguiente se agudiza aún más el fenómeno descrito, puesto que tales programas están diseñados para entregarle todo el ancho de banda posible a un solo usuario, aunque tenga que afectar al resto.

Con el objetivo de asignar un ancho de banda equitativo por usuarios y subredes, se empleó el mecanismo llamado *delay pool*, propio del software Squid-Cache. Este mecanismo es uno de los más útiles pues posibilita manipular el ancho de banda individual que se les asigna a los usuarios que se conectan simultáneamente. Las restricciones sobre los usuarios y subredes son manipuladas a través de los conceptos de *pool*, *bucket* y *class* [LUYER, 2009] que se describen a continuación:

Pool: representa una colección de grupos de *bucket* de un tipo de clase determinada.

Bucket: representa un cubo de asignación de tráfico que se repone a un ritmo determinado, y produce una espera de la solicitud cuando este cubo se encuentra vacío.

Class: representa la clase o nivel en que son aplicados los *bucket* en una *pool*.

Class 1: un *bucket* unificado a nivel de red.

Class 2: un *bucket* unificado a nivel de red y uno individual para cada host.

Class 3: un *bucket* unificado a nivel de red, 255 *buckets* para cada subred y uno individual para cada host.

Aunque existen otras clases de *pool* [FLICKENGER, 2006] [LUYER, 2009], por las características de este caso solo se aplicaron las de clase 2. A cada *pool* debe configurársele de forma eficiente los parámetros o tamaño que poseerán sus grupos de *bucket* según la clase en la que está definida.

## Parámetros de Delay Pool

Cada *pool* contiene parámetros que se definen de la siguiente forma:

$$\text{delay\_parameters} \quad \text{pool} \quad V/V_{\max}$$

*delay\_parameters*: etiqueta que indica que se desea configurar los parámetros de *bucket* para una *pool* determinada.

*Pool*: indica el número de la *pool* a la que se aplicarán los parámetros de *bucket*.

*V*: indica la velocidad en bytes/segundos que tendrá el *bucket*.

*Vmax*: indica la velocidad instantánea máxima en bytes/segundos que tendrá el *bucket* en un momento determinado.

Los parámetros de *bucket* deben ser especificados para cada nivel de la clase, o sea, si la *delay pool* es de clase 3, entonces habrá 3 pares de valores de *V* y *Vmax* en la línea de configuración de parámetros. El primer par para los *bucket* de red, el segundo par para los de subred, y el tercer par para los individuales por PC, según la definición de la clase 3. El valor -1 en los parámetros, indica la asignación de la máxima velocidad disponible.

$$\text{delay\_parameters} \quad 1 \quad -1/-1 \quad 10240/20480$$

El ejemplo anterior indica la configuración de los parámetros para la *delay pool* número 1, de clase 2 puesto que contiene dos pares de valores de *V* y *Vmax*. El primer par define un *bucket* de red de valor -1 indicando que se usará todo el ancho del canal, y el segundo par define un *bucket* por usuario de tamaño de 10 Kbytes/s con valores máximos posibles de 20 Kbytes/s.

Para la medición del ancho de banda consumido por cada usuario se empleó el software squish, módulo que se integra al Squid Caché, ver Figura 1. De los datos provistos por el software se emplearon: la cantidad de usuarios conectados y el ancho de banda consumido promedio.

	<b>Curr. Speed</b>	<b>Avg. Speed</b>	<b>Size</b>	<b>Time</b>
<b>30 users and 214 connections @ 55.48/34.19 KB/s (CURR/AVG)</b>				
	0.27 KB/s	0.27 KB/s	12 Kb	47s
			0 b	5s
	0.27 KB/s	0.27 KB/s		
	1.65 KB/s	1.44 KB/s	43 Kb	30s
	0.22 KB/s	0.21 KB/s	16 Kb	1m 20s
	0.21 KB/s	0.23 KB/s	18 Kb	1m 22s
	0.28 KB/s	0.18 KB/s	15 Kb	1m 25s
	0.47 KB/s	0.46 KB/s	14 Kb	31s
		0.07 KB/s	11 Kb	2m 40s
	2.82 KB/s	2.59 KB/s		

Figura 1: Módulo Squish empleado para monitorear el uso del canal

El ancho de banda actual destinado para la navegación web es de 32 Kbytes/s, aunque puede llegar a los 40Kbytes/s en caso de no usarse el canal dedicado para el correo, e incluso tener picos de velocidad mayores puesto que el canal realmente posee un enlace a 1.5 Mbit/s en su totalidad. Los datos tomados por el Squish durante un período de monitoreo sin aplicar las *delay pools* fueron los siguientes:

Tabla 1: Consumo de Ancho de Banda sin Delay Pools.

No. Usuarios	Uso Promedio del Canal (Kbytes/s)
Menos de 10	29,01
10-20	30,08
20-30	35,56
30-40	35,24
Más de 40	36,75

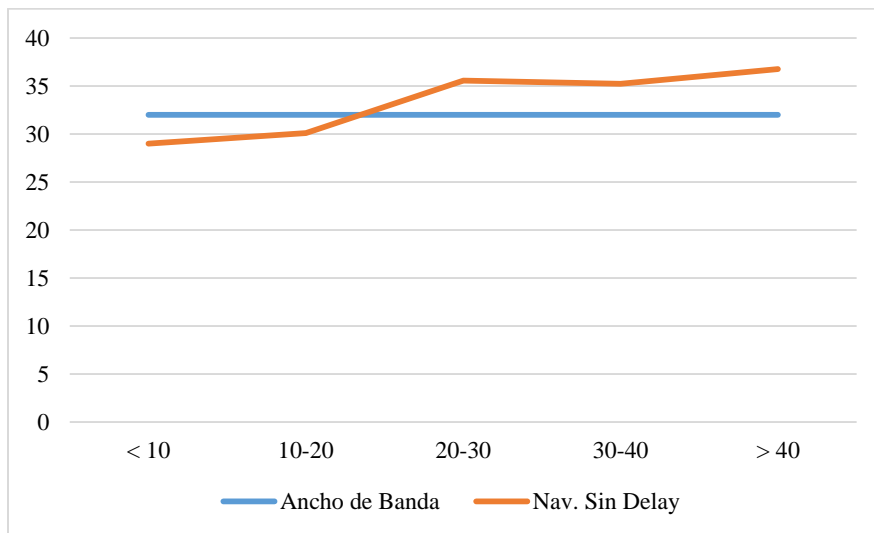


Figura 2: Consumo de ancho de banda sin delay pools.

Los datos recopilados muestran que sin el mecanismo de *delay pool* el ancho es eficientemente bien usado, incluyendo parte del canal destinado para correo, es por eso que en promedio la gráfica sale por encima de los 32 Kbytes/s. Esto solo nos brinda una muestra el consumo de todos los usuarios de la red, pero no el individual, donde la competencia hace que unos queden sin ser atendidos.

Como primera instancia para repartir equitativamente el canal entre todos los usuarios, se determinó la moda de usuarios que son atendidos en horario laboral y no laboral.

$$\text{Usuarios horario laboral (Uhl)} = 40$$

$$\text{Usuarios horario no laboral (UhnL)} = 12$$

Si se pretende repartir equitativamente el canal por usuarios los parámetros para definir el tamaño de los *buckets* asignados a los usuarios sería:

$$\text{Tamaño del bucket (Tb)} = \text{Ancho del Canal (Ac)} / \text{Cantidad de Usuarios (U)}$$

$$Tb = 32 \text{ Kbytes/s} / 40$$

$$Tb = 0,8 \text{ Kbytes/s}$$

Este tamaño de *bucket* sería insuficiente para la navegación, una página de tamaño promedio de 500 Kbytes (según promedio de sitios accedidos desde la universidad), tardaría unos 10,4 minutos haciendo la navegación imposible. Además los usuarios nunca llegan a emplear todo el canal constantemente, sino más bien durante el tiempo que demora responder la petición por lo que el resto del tiempo queda desocupado el canal, puesto que es imposible que todas las peticiones coincidan en el mismo tiempo. Teniendo esto en

cuenta se estima teóricamente el canal al doble, por lo que los parámetros quedarían de la siguiente forma:

$$Tb = 2 * Ac / U$$

$$Tb = 2 * 32 \text{ Kbytes/s} / 40$$

$$Tb = 1.6 \text{ Kbytes/s}$$

Configurando los parámetros de *delay pool* para un bucket de tamaño 1.6 Kbytes/s se obtuvo el siguiente resultado:

Tabla 2: Consumo de Ancho de Banda con Delay Pools, Caso 1.

No. Usuarios	Uso Promedio del Canal
Menos de 10	20,6
10-20	23,95
20-30	25,49
30-40	24,3
Más de 40	26,14

Con estos parámetros si bien se puede evitar que los usuarios consuman el ancho de banda asignado para otros usuarios, se termina desperdiciando parte del canal total. Las restricciones de *bucket* dividen el canal por usuarios, pero si uno de ellos no lo usa en su capacidad entonces se desperdicia. La solución a este problema se encuentra aumentando el tamaño de los *buckets*, aunque si se aumentan mucho entonces se asemeja al acceso por contienda sin las *delay pools*. Un aumento de la eficiencia en el empleo del canal se logra aumentando el factor de multiplicidad en 3, o sea estimando el ancho del canal al triple y calculando la *Vmax* por la menor cantidad de usuarios conectados promedio, que en este caso es de 10 personas.

$$Tb = 3 * Ac / U$$

$$Tb = 3 * 32 \text{ Kbytes/s} / 40$$

$$Tb = 2,4 \text{ Kbytes/s}$$

$$Vmax = Ac / Umin$$

$$Vmax = 32 \text{ Kbytes/s} / 10$$

$$Vmax = 3,2 \text{ Kbytes/s}$$

Con esto queda la configuración en el Squid Caché de la siguiente forma:

```
delay_parameters 1 -1/-1 2458/ 3277
```

Los resultados obtenidos con dicha configuración fueron los siguientes:

Tabla 3: Consumo de Ancho de Banda con Delay Pools, Caso 2.

No. Usuarios	Uso Promedio del Canal
Menos de 10	30,05
10-20	29,97
20-30	30,94
30-40	30,05
Más de 40	31,45

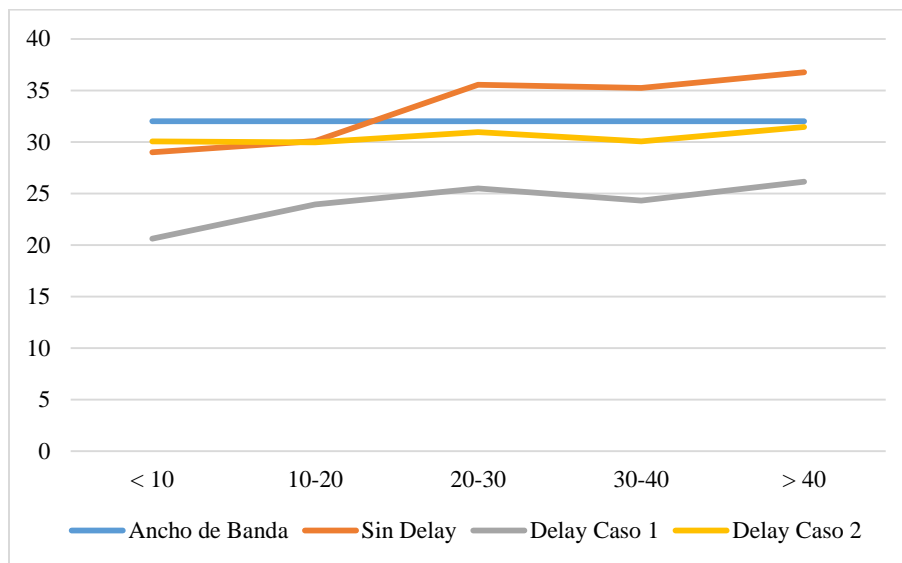


Figura 3: Consumo de ancho de banda por variantes.



## Conclusiones

El ancho de banda de manera general se consume mejor al no haber restricciones de demora con las *delay pools*, pero la navegación termina siendo por contienda, beneficiándose algunos y dejando a otros sin acceso. El uso de *delay pools* requiere una correcta configuración de sus parámetros, de lo contrario se puede desaprovechar el canal como en el caso 1, y con una buena configuración de ellos se puede lograr una distribución más equitativa del ancho de banda al mismo tiempo que se emplea con eficiencia. Aumentando el tamaño de los *buckets* se tiende a un acceso por contienda, y disminuyéndolos a desaprovechar el canal.

Otras alternativas que se pueden emplear para favorecer la navegación es configurando *delay pools* para archivos de menor interés como imágenes, ejecutables, audio y video, asignándole menor ancho de banda general y por usuario.

El empleo de los parámetros para la configuración de las *delay pools* del caso 2, permitió una navegación por usuarios más equitativa al mismo tiempo que se realiza un uso de canal más eficiente, logrando una mayor atención de usuarios en tiempos relativamente aceptables. Si bien las *delay pools* no hacen milagros con un ancho de banda ineficiente, su correcto uso puede ayudar a mejorar la calidad del servicio.

## Bibliografía.

FLICKENGER R. *How to Accelerate Your Internet: A Practical Guide to Bandwidth Management and Optimisation Using Open Source Software*, Broadband communication systems, 2006, p 190, 298.

LUYER D. *SQUID Frequently Asked Questions*, [on-line], 2009, [consultado: noviembre de 2013], Disponible en: <http://pozi.dsic.upv.es/swe/soft/squid/squid-faq.pdf> ó <http://wiki.squid-cache.org/Features/DelayPools>.

*Squid, configuration directive delay\_parameters*, [on-line], [consultado: noviembre de 2013], [http://www.squid-cache.org/Doc/config/delay\\_parameters/](http://www.squid-cache.org/Doc/config/delay_parameters/).