



Acelerar las descargas con Trickle

# CONTROL DEL TRÁFICO

¿Sufre nuestra conexión a Internet el castigo de un exceso de descargas? Si es el caso, deberíamos probar Trickle, un sencillo programa que nos ofrece un control más preciso en el tráfico de nuestra conexión a la red. **POR OLIVER FROMMEL**

Las descargas de grandes ficheros pueden dejar nuestra conexión prácticamente inutilizable. Algo parecido a lo que ya ocurría en épocas pasadas, en las que el usuario tenía que esperar segundos hasta ver el eco de lo que pulsaba en pantalla.

TCP/IP necesita suficiente capacidad en ambas direcciones. Sin importar si estamos descargando o subiendo información, los paquetes viajan en ambas direcciones. El dispositivo en el extremo de la recepción de una descarga tiene que confirmar la recepción de cada paquete que recibe, y obviamente esta confirmación viaja en dirección contraria hacia la fuente.

## Moldeado de Tráfico Fácil

Para usar el ancho de banda de manera más eficiente, es importante tener un control más preciso sobre los paquetes que entran y salen de nuestra máquina. Linux tiene ciertas funcionalidades de administración de colas que proporcionan una amplia gama de métodos de traffic shaping, basados en diversos algoritmos. El traffic shaping es una técnica para controlar el tráfico entrante y

saliente y hacer un uso óptimo de la capacidad de la conexión. No hace falta decirlo: el sistema es tan complicado como su nombre (véase [1]). La herramienta Trickle [2] soluciona el problema proporcionando un comando sencillo con sólo unas pocas opciones. Trickle normalmente es sencillo de compilar. Requiere sólo Libevent [3] y algunas otras librerías estándar.

## Sin privilegios

Trickle ni siquiera necesita privilegios de administrador para controlar el tráfico. Normalmente, Glibc proporciona aplicaciones con ciertas funciones de red que les permiten establecer conexiones con Internet, transferir archivos y demás. Cuando Trickle arranca, usa la variable de entorno `LD_PRELOAD` para cargar su propia librería dinámica, que incluye funciones de red. Funcionan igual que las originales, pero que además registran los flujos de tráfico. Trickle funciona con la mayoría de los programas, salvo unas pocas excepciones. Si encontramos dificultades para poner Trickle en marcha con nuestra aplicación favorita, deberíamos

echarle un vistazo al cuadro “Restricciones”.

Algunas aplicaciones no usan el mecanismo por defecto. Para someter a un programa al control de ancho de banda, necesitamos arrancarlo con el comando `trickle`. El parámetro `-d` nos permite especificar un límite para la tasa de descarga:

```
trickle -d 50 ↵
ftp ftp.redhat.com
```

Esto le indica a Trickle que asegure que el flujo de descarga de la conexión `FTP` no

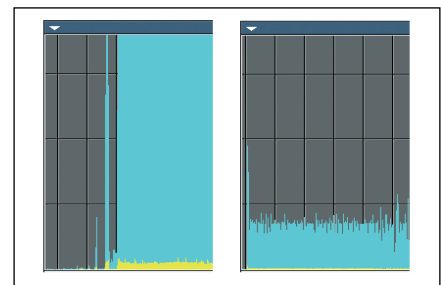


Figura 1: Sin el control de Trickle, la descarga FTP consume todo el ancho de banda disponible. Si fijamos un límite, tendremos capacidad suficiente para otras aplicaciones de red.

excederá los 50 Kbps de ancho de banda de media. La opción `-u` hace lo mismo para el flujo de subida. Otros programas FTP que se ejecuten al mismo tiempo no se ven afectados por Trickle y podrían continuar apoderándose de la conexión. Para proporcionar un control más preciso, tendremos que habilitar Trickle para que controle todos los programas que envíen o reciban información por Internet. El demonio de Trickle, *trickled*, se ocupa de esto, registrando todas las conexiones de Trickle y controlando el ancho de banda total consumido.

## Controles Globales

De nuevo, el demonio soporta los dos sencillos parámetros que comentamos anteriormente, `-d` y `-u`, pero en este caso referidos al ancho de banda total:

```
trickled -d 50 -u 5 -f -N 5
```

Este comando fija la tasa de descarga a 50 Kbps y la de subida a 5 Kbps. El parámetro `-f` le indica al demonio que se ejecute en segundo plano. `-N 5` hace que el demonio refresque las estadísticas cada 5 segundos. Si omitimos el parámetro `-d`, Trickle fijará las tasas de subida y bajada a 10 Kbps. Podemos guardar configuraciones de manera permanente en el archivo `/etc/trickled.conf`, o fijar la opción `-c` y especificar un nombre de archivo.

Trickle programa la comunicación con el demonio al iniciarse (normalmente a

## Restricciones

Debido a que el mecanismo asociado a `LD_PRELOAD` usa librerías dinámicas, Trickle no funcionará con los denominados binarios estáticos que no usan Glibc. El comando `ldd` nos indicará el tipo de programa que le estamos pidiendo a Trickle que controle. Al mismo tiempo, `ldd` nos ofrece una lista de librerías dinámicas:

```
ldd /usr/bin/wget | grep libc.so
libc.so.6 =>
/lib/libc.so.6 (0x00add000).
```

Si `ldd` no nos muestra ningún resultado, podemos suponer que el candidato es un programa estático. Podemos verificarlo tecleando el comando `archivo /usr/bin/wget`. Además de esto, Trickle no puede tratar programas SUID, que desactivan el mecanismo asociado a `LD_PRELOAD` por razones de seguridad.

través de un **socket Unix** llamado `/tmp/.trickled.sock`), y controla los programas cliente en función de nuestras reglas. Si la conexión con el servidor se interrumpe, Trickle sigue funcionando, pero sin respetar los límites. Si el servidor funciona con un límite de descarga de 10 Kbps, al teclear `trickle wget http://www.w3.org` se respetará el límite. Pero si desconectamos el servidor mientras se está efectuando la descarga, la tasa de descarga se incrementará.

## Suavizado

Podemos usar el archivo de configuración `/etc/trickled.conf` para definir las prioridades de los servicios de manera individual. Los valores menores representan prioridades mayores. Internamente, Trickle trata el orden de los paquetes en una cola. Este método significa que podemos conseguir velocidades de descarga bastante buenas y aún así poder usar sesiones `ssh` sin problemas. Veamos un sencillo ejemplo de un archivo de configuración:

```
[ssh]
Priority = 1
[www]
Priority = 8
```

La documentación también recomienda los parámetros `Time-Smoothing` y `Length-Smoothing` para evitar fluctuaciones en la tasas de transferencia. Al contrario que con las opciones en línea de comandos, `-d` y `-l`, podemos fijar los parámetros en el archivo de configuración de manera individual para cada servicio (SSH, FTP, WWW,...):

```
[ssh]
Priority = 1
Time-Smoothing = 0.1
Length-Smoothing = 2
[www]
Priority = 8
Time-Smoothing = 5
Length-Smoothing = 20
```

Estos valores definen el tiempo y la longitud de normalización que Trickle aplica al programa que está controlando. Para transferencias de archivos grandes se recomiendan los valores mayores, mientras que las aplicaciones interactivas deberían usar valores menores. Ninguno de estos valores tuvo demasiado efecto en las pruebas, pero podemos experimentar por nuestra cuenta.

## Límites

Existen ciertos límites a lo que puede hacer un sistema de estas características. No importa si el kernel se está ejecutando en espacio de usuario, ni la influencia de los búfers (que se fijan para mantener unos pocos paquetes), ni tampoco de la naturaleza dinámica de las redes. Trickle calcula las tasas sobre un periodo específico de tiempo. Como la tasa comienza con un valor bajo y comienza a crecer con el tiempo, la herramienta debería llegar más allá de la marca y subir por encima del valor que tenemos como objetivo.

`-w` especifica el número de bytes en el cual Trickle debería intentar evitar el tráfico a ráfagas, aunque debemos decir que esta configuración tuvo poco efecto en nuestras pruebas. La diferencia con el valor por defecto de 512 bytes era apenas perceptible. Dicho esto, Trickle aún hace un buen trabajo sin necesidad de un ajuste fino.

## Idóneo Para Casa

Trickle está diseñado principalmente para proporcionar al usuario un mayor control de los flujos de tráfico generados o recibidos por aplicaciones individuales. Un uso multiusuario es concebible, pero implicaría escribir scripts por encima para controlar todas las aplicaciones involucradas, por lo que puede que sea preferible una solución basada en QDisc (véase [1]). El programa puede tratar sólo conexiones TCP, las típicas en la mayoría de servicios con alto volumen de transferencia. La mayoría de los usuarios podrán vivir con el hecho de que Trickle no puede tratar tráfico DNS, por ejemplo. La herramienta de `traffic shaping` Trickle es definitivamente una buena solución para los lectores que necesitan un método rápido y sencillo de mantener una sesión interactiva mientras que se transfieren grandes cantidades de información. ■

## Recursos

- [1] CÓMO "Advanced Routing and Traffic Control": <http://lartc.org>
- [2] Trickle: <http://monkey.org/~maris/pages/?page=trickle>
- [3] Libevent: <http://www.monkey.org/~provos/libevent>