

mldonkey en Debian Woody

Ignacio Barrancos Martínez
ignacio@adesx.com

Contenido

1.Introducción.....	1
2.Instalación en Debian Woody.....	1
2.1.Configuración de mldonkey.....	3
2.2.Configuración del router o cortafuegos.....	5
2.3.Configuración de Apache.....	5

1. Introducción

MLdonkey es un cliente p2p multiplataforma y multired escrito en Objective-Caml (ocaml), un lenguaje declarativo que se ejecuta en la mayoría de los Sistemas Operativos.

Inicialmente, fue el primer cliente OpenSource que podía acceder a la [red eDonkey](#). Para descifrar el protocolo se usó ingeniería inversa con el programa-sniffer [pandora](#). Actualmente, junto a eDonkey, se soportan otras redes p2p como [Overnet](#), [Bittorrent](#), [Gnutella](#), [Gnutella2](#), [Fastrack](#) (Kazaa), [Soulseek](#), [Direct-Connect](#) y [Openmap](#). Las redes se pueden habilitar/deshabilitar para simultanear las búsquedas de archivos, pero estos sólo se descargarán de un tipo de red, aunque de múltiples clientes en paralelo.

Mldonkey es ejecuta en nuestro equipo como un demonio o servicio, pudiendo controlarlo a través de múltiples interfaces:

- La forma más simple sería mediante telnet al puerto 4000,
- La más interesante a través del navegador web (<http://equipo:4080>) ,
- O biné a través de un cliente GTK.

Todas estas interfaces se pueden usar de forma local o remota, una vez se hayan deshabilitado o configurado las restricciones de seguridad.

En este documento se describe cómo compilarlo e instalarlo sobre Debian GNU/Linux, habilitando la interfaz http para poder controlarlo de forma remota, sin necesidad de estar en casa para añadir nuevas descargas, o ver el estado de las mismas.

2. Instalación en Debian Woody

Lo primero que se debe hacer es **descargar la versión de mldonkey** que se desea instalar: El sitio oficial es <http://www.nongnu.org/mldonkey/>, y las versiones se pueden descargar de esta [otra url](#). Para poder compilarlo en nuestro equipo será **necesario tener instalados los paquetes zlib y zlib-dev**. En Woody lo haremos mediante:

```
apt-get install zlib1g zlib1g-dev
```

Una vez descargada la versión de mldonkey, se descomprimirá y ejecutará el archivo `./configure`.

```
tar -xzvf mldonkey-2.5.21.tar.gz
cd mldonkey-2.5.21
./configure
```

Al hacerlo, el script detectará que no se tiene instalada la versión 3.07 de ocaml, y nos preguntará si deseamos descargarla e instalarla: Le diremos que sí (y).

```
Do you want this script to try to download and install ocaml
LOCALLY in mldonkey directory ?
y
```

Tardará unos minutos, antes de terminar. Luego se ejecutará `make` ...

```
make
```

Este comando compilará mldonkey y nos dejará un ejecutable en el directorio actual llamado `mlnet`: es lo único que se necesitará.

Una vez se tiene el ejecutable, convendría crear una cuenta de usuario, para ejecutar el demonio, en vez de que se ejecute como `root`: para ello teclee ...

```
adduser donkey
mkdir ~donkey/mlDonkey
cp mlnet ~donkey/mlDonkey
```

Así, en el directorio `mlDonkey` del home del usuario `donkey`, se tendrán todos los ficheros de configuración y descargas del cliente `p2p`. Si se cambia de ordenador, se reinstala o se quiere instalar en otro equipo, bastará con copiar este directorio al nuevo destino sin necesidad de volver a compilar, y sin perder el estado actual de las descargas ni la configuración.

El siguiente paso será **arrancar el demonio para que se generen los ficheros de configuración iniciales**, además de cambiar la contraseña del administrador del cliente, y añadir al menos un usuario. Esto se hará desde el cliente `telnet`, para lo cual procederemos de la siguiente forma:

1. Arrancar el servicio:

```
cd ~donkey/mlDonkey
./mlnet -daemon
```

2. Conectarse vía `telnet`:

```
telnet localhost 4000
```

3. Habilitar los clientes de otras redes:

```
> networks
```

Mostrará el listado de las redes y su configuración:

```
Networks:
 7 Donkey           Enabled
 6 BitTorrent       Enabled
 3 G2                Disabled
 4 Fasttrack        Disabled
 2 Gnutella          Disabled
 5 FileTP            Enabled
```

Se habilitarán mediante el comando enable, para ello...

```
> enable 3
> enable 4
> enable 2
```

4. Cambiar la contraseña del administrador:

```
> add_user admin mi_passwd
```

donde mi_passwd se sustituirá por la contraseña que queramos ponerle.

5. Añadir un nuevo usuario:

```
> add_user mihermano su_password
```

donde mihermano será el login del nuevo usuario.

6. Guardar los cambios y salir:

```
> save
>q
```

Si ahora listamos el contenido del directorio actual veremos cómo se han creado varios archivos .ini y algún que otro directorio. Destacan

- **El archivo download.ini**, donde se almacena la configuración de nuestro cliente, reserva del ancho de banda, puertos de salida, etc...
- **El directorio incoming**, donde se almacenarán los archivos descargados
- **El directorio temp**, que contiene los archivos que actualmente se están descargando, pero su nombre será un hash, y no podremos identificarlos fácilmente.

2.1. Configuración de mldonkey

Una vez se esté ejecutando mldonkey, observaremos cómo el ancho de banda para navegar por Internet disminuye considerablemente, así como la velocidad de nuestro sistema Linux, llegándose incluso a colgar.

Para evitarlo, es recomendable limitar el consumo de recursos, modificando el archivo de configuración a mano, puesto que mi experiencia con versiones diferentes me dice, que si esto se hace a través de alguna interfaz, los cambios no se aplicarán. Para poder hacer esto, seguir los siguientes pasos:

1. Parar el demonio mlnet, como usuario root:

```
killall -9 mlnet
```

2. Editar el archivo ~donkey/mlDonkey/downloads.ini, y limitar los siguientes parámetros:

- max_hard_download_rate = 17

Con ello limitaremos el ancho de banda de la descarga a 17 KB/s. Por defecto viene configurado a 50, muy superior al ancho de banda disponible para los usuarios de ADSL.

- max_opened_connections = 175

Por defecto, inicializada a 200.

- force_client_ip = false

Si usamos ip dinámica, a true, si nuestra ip es estática.

3. Guardar los cambios y salir.

Ahora podríamos crear un script para arrancar demonio mlnet y que se ejecutara como el usuario donkey en vez de como el root. Para ello, crear un archivo en /etc/init.d/mldonkey-daemon con el siguiente contenido:

```
#!/bin/sh

#-Definición de los colores ....
RES_COL=60
MOVE_TO_COL="echo -en \\033[${RES_COL}G"
SETCOLOR_SUCCESS="echo -en \\033[1;32m"
SETCOLOR_FAILURE="echo -en \\033[1;31m"
SETCOLOR_WARNING="echo -en \\033[1;33m"
SETCOLOR_NORMAL="echo -en \\033[0;39m"

#- Para mostrar el OK con colores como en RedHat
echo_success() {
    $MOVE_TO_COL
    echo -n "[ "
    $SETCOLOR_SUCCESS
    echo -n "$OK"
    $SETCOLOR_NORMAL
    echo -n "]"
    echo -ne "\r\n"
    return 0
}

#- Para mostrar el KO con colores como en RedHat
echo_failure() {
    $MOVE_TO_COL
    echo -n "["
    $SETCOLOR_FAILURE
    echo -n "$FAILED"
    $SETCOLOR_NORMAL
    echo -n "]"
    echo -ne "\r\n"
    return 1
}

#-Comprobamos que sólo el root pueda ejecutar el script
if [ "$USER" != "root" ]; then
    $SETCOLOR_FAILURE
    echo -n "Sólo el root puede realizar esta operación"
    $SETCOLOR_NORMAL
    echo -ne "\r\n"
    exit 1;
fi

#- Usuario que ejecutará mlDonkey.
RunUser=donkey

case "$1" in
    start)
        echo -n "Arrancando mlDonkey..."
        cmd='cd /home/donkey/mlDonkey && ./mlnet -daemon 2>&1 > /dev/null &'
        su $RunUser -c "$cmd" > /dev/null 2>&1
        RETVAL=$?
        [ $RETVAL -ne 0 ] && echo_failure
        [ $RETVAL -eq 0 ] && echo_success
        ;;
    stop)
        echo -n "Parando mlDonkey..."
        /usr/bin/wget --http-user=admin \
            --http-passwd=PONER_PASSWORD \
            http://localhost:4080/submit?q=kill \
            > /dev/null 2>&1
    esac
```

```

    RETVAL=0
    echo_success
    ;;
status)
    RETVAL=0
    ;;
restart|reload)
    $0 stop
    $0 start
    RETVAL=$?
    ;;
*)
    echo "Uso: $0 {start|stop|status|restart|reload}"
    exit 1
esac
exit $RETVAL

```

...donde deberemos sustituir `PONER_PASSWORD` por el password que pusimos al usuario admin, cuando nos conectamos mediante telnet. Después se ejecutará...

```
update-rc.d mldonkey-daemon defaults
```

Con lo que quedará configurado para arrancar y parar con el sistema.

2.2. Configuración del router o cortafuegos

Si estamos conectados a Internet mediante un router ADSL, deberemos asegurarnos de que **se tienen re-direccionados los puertos 4661 y 4662** al equipo en el que instalamos mldonkey en nuestra intranet. Para los modelos *Zyxel Prestige 643* que vende telefónica se hará con la opción 15. *SUA Server Setup*, en el menú de administración del router.

Si por el contrario tenemos configurado un cortafuegos, deberemos habilitarle las reglas necesarias para permitir tráfico en estos puertos.

2.3. Configuración de Apache

Lo interesante de usar mldonkey es la posibilidad de poder administrarlo de forma remota a través de Internet, sin necesidad de estar sentados delante de nuestro equipo o en la propia intranet. Ello es posible habilitando una redirección de URLs a Apache, para que determinadas URLs las redirija al demonio mldonkey. Para conseguirlo, añade la siguiente configuración a su `httpd.conf` (*la configuración propuesta es para Apache 2, para Apache 1 es análoga, sólo cambiarán las rutas*):

```

LoadModule cache_module /usr/lib/apache2/modules/mod_cache.so
LoadModule disk_cache_module /usr/lib/apache2/modules/mod_disk_cache.so
LoadModule proxy_module /usr/lib/apache2/modules/mod_proxy.so
LoadModule proxy_http_module /usr/lib/apache2/modules/mod_proxy_http.so
<IfModule mod_proxy.c>
  <Proxy *>
    Order allow,deny
    Allow from all
  </Proxy>
  ProxyVia On
  CacheRoot "/var/cache/apache2/proxy"
  CacheSize 5
  CacheGcInterval 4
  CacheMaxExpire 24
  CacheLastModifiedFactor 0.1
  CacheDefaultExpire 1
  <VirtualHost *>
    DocumentRoot /prj/www #-Poner lo que tú tengas
    #Esto es lo importante para la redirección
    RewriteEngine On
    ProxyRequests on
    RewriteLog /var/log/apache2/rewrite.log
    RewriteLogLevel 9
    RewriteRule /mldonkey/(.*) http://localhost:4080/$1 [P,L]
    ProxyPassReverse /(.* http://localhost:4080/$1

```

```
</VirtualHost>  
</IfModule>
```

Reiniciar el servicio apache, e intentar acceder a la URL `http://ip-mi-servidor/mldonkey/`, ojo que la última barra es importante.