

Radio en Internet:¿Cómo hacer streaming?



Pancake / CatHack!

sábado 29 noviembre 2003.

Poco a poco, vamos viendo cómo Internet se consolida como un importante medio de comunicación; es en este punto donde las radios libres, dignas portadoras de la libertad de expresión, ven la necesidad de volcarse sobre este medio.

Este texto está pensado para todos los interesados en este tema, con el objeto de facilitar la tarea de configuración y la comprensión del streaming.

Pues bien, el streaming es el hecho de emitir en tiempo real algún tipo de medio multimedia (audio/video) a través de un medio informático.

Aquí me basaré principalmente en el streaming de audio sobre software libre, ya que no puedo entender la libertad de expresión sobre un software comercial y de código cerrado, por ejemplo. El streaming de video, ya lo trataré más adelante cuando hayan televisiones libres. ;)

Requisitos

Para hacer streaming es necesario tener:

Servidor de streaming:: realiza la emisión hacia los oyentes

Encoder:: envía la emisión al servidor de streaming

Cliente:: el que lo escucha ;)

Servidores de streaming

Dentro del software libre nos encontramos solamente con una opción (actualmente) de servidor para hacer streaming de audio. Se trata de "IceCast"; también existe el "ShoutCast", que es de código abierto, pero no es software libre, a demás de que su desarrollo depende de una empresa.

Así que no nos preocuparemos más e iremos a configurar directamente el IceCast.

Existen dos versiones:

1 – estable, sólo soporta mp3, funciona muy bien.

2 – en desarrollo, soporta mp3, ogg y relay de otros servidores.

Si realmente queremos montar un sistema de streaming de audio totalmente libre lo tendríamos que hacer sobre **ogg**, ya que **mp3** no es libre y depende de unas patentes muy restrictivas, es decir, el mp3 es ilegal. En

Radio en Internet: ¿Cómo hacer streaming?

cambio ogg es un formato libre y no sufriremos ningún problema moral. ;)

Se recomienda disponer de la versión 2 aunque esté en desarrollo.

Instalación del servidor

En Debian :

```
# apt-get install icecast-server
```

La versión actual de sid del icecast-server es la 1.3.11-4.3, y todas las versiones iguales o anteriores a la 1.3.11 son vulnerables a un ataque de buffer overflow, así que si de verdad no queréis sufrir lo mejor será que os bajéis las sources y las utilicéis de allí.

En cualquier otra... espero que no sea RedHat ;)

Os bajáis las sources y las compiláis

```
$ wget http://www.icecast.org/releases/icecast-1.3.12.tar.gz
# tar xzvf icecast-1.3.12.tar.gz
# ./configure
# make install
```

Es muy recomendable editar ciertos parámetros de compilación:

(./configure)

--with-libwrap : permite crear ACLs

--with-crypt : permite pwd encriptats

(./Makefile)

CFLAGS = -O5 -march=i686 -mcpu=i686 -fstrict-aliasing

(./src/icetypes.h)

SOURCE_BUFFER_SIZE : fer el buffer d'entrada més gran per evitar sotrac

(./src/icecast.h)

OPTIMIZE

SAVE_CPU (suport per threads i semàfors...mutex...)

Según dicen, al activar estas dos opciones se come un 75% menos de CPU :)

Ejecutando el servidor IceCast

Al ejecutarlo podemos pasarle algunas opciones, aunque es mejor definirlas en el fichero de configuración.

-c [filename]: especifica el fichero de configuración.

Radio en Internet:¿Cómo hacer streaming?

- P [port]** : port(s) en los que escuchará (8000,8001 por defecto)
- m [#maxclients]** : número máximo de clientes.
- p [enc_pwd]** : define el password del encoder.
- b**: lanza el demonio en segundo plano.

Fichero de configuración

Por defecto éste es: **/etc/icecast/icecast.conf**

NOTA: Si hemos compilado el servidor con soporte **--lcrpt** tendremos que utilizar la herramienta **"mkpasswd"** que nos dará los passwords encriptados y tendremos que escribir en el fichero de configuración los passwords encriptados y no de texto plano. En Debian, este programa viene junto con el paquete **"whois"** (**apt-get install whois**)

Coged este fichero de configuración y lo adaptáis a vuestro gusto:

```
-----[ icecast.conf ]-----<-----cut here-----
# Pancake Icecast Example...
location Illa Perejil
rp_email contacte@amb.l'admin
server_url www.servidoricecast.org
use_meta_data 0</b> # si ponemos 1 saldrán los datos del stream
streamurl url.siElsSourcesNoEnDefineixenCap.org
streamurllock 1 # las sources no pueden definir ninguna URL
streamtitletemplate título_de_elstream_o_nombre_de_fichero
port 8000
port 8001
#include filename # podemos incluir otros ficheros de configuración.
alias / /default_radio # por defecto cargará esta radio.
#alias /relay_radio http://1.1.1.1:8000/remote_radio # hace un relay :)
# passwords
encoder_password #password-encriptado#
admin_password #password-encriptado#
oper_password #password-encriptado#
touch_freq 5 # cada 5 segundos IceCast toca los directorios (touch)
max_clients 10
```

Radio en Internet:¿Cómo hacer streaming?

```
max_sources 10

max_clients_per_source 10

max_admins 1</b>

logfile /var/log/icecast/icecast.log

#hostname emitir_solo_en_este_host

server_name ip_del_server

force_servername 0 # no hace falta forzar nada, que nos haremos daño :)

reverse_lookups 0 # así es más rápido

# force_servername

console_mode 3 # en segundo plano (0=admin+log, 1=admin, 2=log, 3=bg)

client_timeout 10# segundos que mantiene a los clientes cuando la fuente

# ha desaparecido.

# stats_log (estadísticas)

# statshtml_log (statistics.html)

# stats_time (cada cuanto tiempo actualiza las estadísticas)

# status_time # cada cuanto salen los datos por la consola de admin

accessfile /var/log/icecast/access.log

usagefile /var/log/icecast/usage.log

logdir /var/log/icecast/

#templatedir /dir... # directorio donde se encuentran el stats.html

#http_admin 0 # interficie http?

mount_fallback 0 # si no está la source devuelve un 404

transparent_proxy 0 # si ponemos un 1, hace de proxy; mejor que no ;)

#acl_policy 1 # habilita o deshabilita los ACL. (control de acceso)

throttle 1 # 1MB/s

kick_relays 3 # si hay una source que es un relay, pasados '3'

# segundos (si no tiene clientes) cortará esta conexión.

kick_clients 1 # si no hay source los hecha fuera, sino los redirije

# a otro stream.

relay_reconnect_max -1 # intentará infinitamente hasta reconectar con

# un relay que ha caido.
```

Radio en Internet:¿Cómo hacer streaming?

```
relay_reconnect_time 30 # cada 30 segundos.  
sleep_ratio 0.3 # cuanto más pequeño más CPU consume.  
-----[ icecast.conf ]-----8<-----cut here-----
```

Administración del servidor

Podemos administrar la máquina en tiempo real, a través de una conexión por telnet al puerto 8000 (o al que tengamos configurado).

```
$ telnet 0 8000  
Trying 0...  
Connected to 0.  
Escape character is '^'.
```

Entonces escribimos:

ADMIN [admin_password]

también podemos loguearnos como OPERs que podremos controlar más comandos:

OPER [oper_password]

Aquí podremos introducir diversos comandos, para administrar los alias y hacer cambios de relays, etc... se hace con el comando "alias". Por ejemplo:

ALIAS ADD /pop /cacatua

ALIAS DEL /pop

ALIAS LIST

Crando listas de acceso (ACL), es decir permitir o denegar la entrada a quién queramos:

allow|deny add

Para listar a la gente conectada, podemos utilizar 4 comandos:

"ADMINS" – administradores.

"LISTENERS [hostmask]" – oyentes.

"SOURCES" – los emisores

"LIST" – lo lista TODO

también podemos hacer que un stream se quede grabado en el disco local en un fichero mp3 (u ogg;) con la orden "dump", así:

DUMP /tmp/filename.mp3

Podemos modificar los datos de cada source (mountpoint, url, nom, genere..)

Radio en Internet:¿Cómo hacer streaming?

con la orden "**MODIFY**" así:

MODIFY -m "mountpoint"

-n "nombre"

-u "url"

-d "descripción"

-g "genero"

-p "public"

-b "bitrate"

-P "prioridad" (para el default mount point)

Podemos parar la emisión de un stream con la orden "**PAUSE**" y "**UNPAUSE**" para quitarle la pausa... esto tiene cierta lógica XD

Escoger otro fichero de configuración en el aire: "**REHASH /etc/icecast.conf**"

Al igual que podemos cargar un fichero de configuración en el aire, también podemos cambiar las variables definidas en él, así:

SET max_clients 20

Si llamamos "**SET**" sin parámetros nos listará todas las variables existentes.

Podemos mover los usuarios de una source a otra con la orden "**SELECT**"

Para ver todo lo referente a los logs y las estadísticas tenemos diversos comandos:

UPTIME – tiempo que lleva el servidor funcionando

TAIL – lista los últimos mensajes del log

STATUS – ancho de banda, etc...

DESCRIBE

AUTH – lista las autorizaciones de conexión.

Parar comunicarnos con otros administradores podemos utilizar el comando **TELL**

Y finalmente para salir de la consola escribimos "**QUIT**" :) o si somos unos radicales podemos salir con "**SHUTDOWN missatge**" y entonces el servidor caerá XDD

Relaying

Radio en Internet: ¿Cómo hacer streaming?

Puede ser interesante poder compartir el ancho de banda para realizar streaming, así comunitariamente se puede crear una red de colaboradores para emitir audio de forma no-centralizada. (a ver si tengo tiempo y hago el programa...) Manualmente podemos hacerlo de tres maneras: proxy, importando y exportando:

Proxy:

Si lo habilitamos en el fichero de configuración, todas las peticiones que se hagan a nuestro servidor se irán montando en el aire de otro servidor, creando ALIAS de manera dinámica.

ALIAS ADD / http://remote.server:8000/mount_point

Exportar:

También podemos realizar el paso inverso, es decir, que un servidor envíe el stream hacia otro servidor como si fuese una source más. Lo hacemos así:

RELAY PUSH [options]

Donde las opciones pueden ser:

-p [password] **-u** [url] **-n** [nombre]
-m [mountpoint] **-g** [género] **-P** [public]
-b [bitrate] **-d** [descripción] **-i** [icy o no]

y para desmontar los relays lo podemos hacer con "**RELAY PULL**"

Configuración de clients de stream

Bien, básicamente cuando hablo de clientes de stream, me refiero a los clientes que se conectan al servidor de stream para emitir el audio. Este flujo de audio puede tener dos fuentes básicamente:

- ▶ entrada de línea/micro
- ▶ listado de mp3

Seguramente, la opción más interesante es la de hacer el stream a través de la entrada de línea, ya que podemos entrale al ordenador la salida de una mesa de mezclas, por ejemplo, y realizar todas las mezclas fuera del ordenador de manera que éste quede libre de esta carga y pudiera sufrir lentitud, etc.

Para poner en marcha este modo, necesitaremos una máquina con tarjeta de sonido, una conexión a la red (lógicamente) y tener instalado gnu/linux (digo gnu/linux porque sólo lo he probado con este); lo más seguro es que con BSD, AtheOS, BeOS, etc, también sea aplicable.

Será necesario configurar la tarjeta de sonido, recompilando el kernel para que dé soporte para la entrada de línea/micro. (Las pruebas que he hecho yo han sido realizadas con una CMI-8338, y tuve que deshabilitar el soporte para 5 altavoces ya que 2 de ellos utilizaban la entrada de línea como salida de audio).

Para comprobar si funciona la entrada de audio podemos hacer esto:

Radio en Internet:¿Cómo hacer streaming?

```
$ strings /dev/dsp
```

Si nos dice que no tenemos permisos será necesario entrar como root y darle permisos al usuario.

```
# adduser stream_user audio
```

Volvamos al paso anterior... Si por pantalla nos aparecen varios símbolos de llaves abiertas,querrá decir que funciona la entrada de audio pero que hay silencio; en cambio, si salen símbolos aleatorios querrá decir que la entrada funciona correctamente y no hay silencio en la entrada.

A pesar de que la entrada de audio funcione y podamos oír el echo por los altavoces, no querrá decir que todo está bien; revisemos en la tabla de mezclas que la entrada de línea está activada como "R"ecord, porque sino, por más que entre no permitirá la grabación.

```
$ aumix -l R
```

Así activamos el soporte para "R"ecord de la tarjeta de sonido, pero también podemos hacerlo "gráficamente" si cargamos el aumix sin parámetros, el espacio nos permitirá cambiar esta opción, al igual que con el "rexima", y supongo que con cualquier otra mesa de mezclas por *nix.

Ahora ya tenemos preparada la entrada de audio para realizar el streaming; sólo hará falta configurar el programa que envíe este flujo de audio hacia el servidor.

Subida del stream

Existen diversos programas para enviar el stream hacia el servidor IceCast. Veamos algunos.

mp3 ogg entrada

DarkIce V V línea/micro

IceS V X ficheros (playlist)

MuSE V X line/mic/ficheros

Liveice V X line/mic/ficheros

Como se puede ver, para las necesidades que tenemos, DarkIce es el más interesante para nosotros, ya que es muy estable, no requiere mucha complicación de configuración y permite tanto ogg como mp3, y hará falta tener una máquina dedicada para hacer esta tarea, ya que sólo perdemos 3KB/s de conexión y la entrada de línea, nada más :) (bueno, estaría bien tener el proceso con suficiente prioridad y con iptables asegurar el ancho de banda fijo).

El otro streamer interesante es el MuSe (spam4jaromil! ;) ya que permite realizar mezclas de hasta 6 canales (micro/line-in, ficheros...) y realizar las mezclas en tiempo real; también nos permite cambiar el bitrate en el aire, sin tener que cortar la conexión y volver a realizarla.

A pesar de que sea tan maravilloso :,), la versión actual (0.7.1) es muy inestable y segmenta con facilidad, así que de momento no lo recomiendo.

Así queeee...

Configuración del DarkIce

Radio en Internet:¿Cómo hacer streaming?

Primero tendremos que bajárnoslo:

```
$wget http://prdownloads.sourceforge.net/darkice/darkice-0.9.1.tar.gz
```

Si queremos soporte para mp3 (cosa que es recomendable) necesitaremos el **Lame**:

```
$ wget http://prdownloads.sourceforge.net/darkice/lame-3.91.tar.gz
```

y si queremos para ogg, pues necesitaremos las **libogg** y las **libvorbis**

```
$ wget http://www.vorbis.com/files/rc3/unix/libvorbis-1.0rc3.tar.gz;\nwget http://www.vorbis.com/files/rc3/unix/libogg-1.0rc3.tar.gz
```

Descomprimos y compilamos...

Ok, ahora nos queda un binario en **src/darkice**, y el único parámetro de entrada es

"**-c config.file**" (a parte del verbose=level). Así que editamos un fichero nuevo y lo pasaremos como un fichero de configuración al DarkIce.

He escrito uno muy claro y fácil de modificar; copiadlo y adaptadlo a vuestras necesidades.

```
----[ darkice.conf ]-----8<----cut here-----\n\n# Pancake DarkIce Configuration.\n[general]\n# Segundos teniendo en marcha el darkice? 0=siempre.\nduration = 0\n# Segundos de buffer de entrada de la tarjeta de sonido.\nbufferSecs = 5\n[input]\ndevice = /dev/dsp\n# Puede ser 44100,22050,11025\nsampleRate = 22050\n# 8 || 16\nbitsPerSample = 16\n# 1=mono 2=stereo\nchannel = 1\n\n# Configuración para conectar a servidores IceCast.\n[icecast-0]\n# (El zero puede ser de 0 a 7 (8 salidas)\n\n# bitrate to encode\n\nbitrate = 32\n\n# sampleRate = 22050 # default from input section\n\nserver = 192.168.0.2\n\nport = 8000\n\npassword = ep\n\nmountPoint = pop.mp3 # the mount point name.\n\nname = FreeRadio\n\ndescription = FreeRadioAgainstTheSystem
```

Radio en Internet: ¿Cómo hacer streaming?

```
url = http://cathack.hn.org/new

genre = punk/oi!

public = yes # could be also 'no'

#remoteDumpFile=remote_file_name

#localDumpFile = local_file_name

# Opciones del encoder:

# lowpass (-1) deshabilitado, (0) encoder defaults, (#) lo pasa al lame.

# control de agudos y graves

lowpass = -1

highpass = 0

# Configuración para servidores IceCast2

[icecast2-0]

# [icecast2-1] # almost 8 (0-7) outputs.

# ogg vorbis r00lz ;)

format = vorbis

bitrate = 96

# quality : from 0.0 to 1.0

quality = 0.7

server = 192.168.0.2

port = 8000

password = ep

mountPoint = pop.ogg

# -- opcional -- # (mirar parámetros más arriba)

# sampleRate

# name

# description

# url

# genre

# public

# localDumpFile
```

Radio en Internet: ¿Cómo hacer streaming?

```
# Configuración para servidores Shoutcast

[shoutcast-0]

# hasta 8 servidores shoutcast (0-7)

bitrate = 96

server = 192.168.0.2

port = 8000

password = ep

# -- opcional -- # (tampoco hacen falta muchas explicaciones)

# sampleRate

# name

# url

# genre

# public

# irc

# aim

# icq

# lowpass

# highpass

# localDumpFile

# Grabación local de la entrada de audio.

[file-0]

# 0 puede ser de 0-7

format = mp3

# format = vorbis

bitrate = 96

fileName = local_file_name_output.ogg|mp3

# -- opcional -- #

# sampleRate

# lowpass

# highpass
```

```
-----[ darkice.conf ]-----8<-----cut here-----
```

Como ya he comentado más arriba, el otro streamer que nos interesa es el LiveIce, que viene empaquetado por Debian y permite la emisión de audio entrado por la entrada de línea.

LiveIce

Streamer de ficheros y/o de entrada de línea, soporta diversos encoders de mp3 (L3,LAME,AJ,SCREAM,GOGO,XING); su funcionamiento es parecido al del DarkIce, es decir en un fichero de configuración, aunque éste tiene soporte de hasta 2 canales para hacer mezclas en modo text x (con -M (mixer mode) que nos permite cambiar las velocidades, volúmenes, cambio de ecación, etc. de los dos canales. También lleva un front-end con tcl/tk para hacer un fichero de configuración básico "\$liveiceconfigure.tk".

<http://star.arm.ac.uk/spm/software/liveice.tar.gz>

```
-----[ liveice.conf ]-----8<-----cut here-----

# LiveIce pancake example ...

SERVER 192.168.0.2

PORT 8000

NAME pancake

GENRE pancake

URL cathack.hn.org/new

PUBLIC 1

X_AUDIOCAST_LOGIN

PASSWORD ep

SAMPLE_RATE 22050

MONO

SOUNDCARD

BITRATE 32000

HALF_DUPLEX

USE_LAME3

VBR_QUALITY 1

NO_MIXER

#PLAYLIST playlist

MIX_CONTROL_AUTOMATIC

-----[ liveice.conf ]-----8<-----cut here-----
```

Otros streamers

Radio en Internet: ¿Cómo hacer streaming?

IceS: Según dicen, no recomiendan utilizarlo si no estás muy seguro de lo que haces XD. Simplemente le pasas un listado de mp3 y él los reproduce hacia un servidor IceCast

<http://www.icecast.org>

MuSE: Multiple Stream Engine, tal como su nombre indica, nos permite mezclar diversas entradas de audio (otros servidores de stream, ficheros mp3 y listas de mp3 (.pl (playlist)) y la entrada de micro; supongo que pronto lo adaptarán para que soporte ogg. Osea, que soporta 6 entradas de ficheros y la de micro, subiendo/bajando el volumen, etc. Además permite cambiar el bitrate del stream en el aire, sin tener que reiniciar las conexiones. El único defecto que le encuentro es que actualmente sufre demasiados SegFaults

<http://muse.dyne.org>

Calidades y anchos de banda

Es muy importante tener en cuenta la calidad en la que emitimos, ya que de este punto dependerá quién podrá escucharnos, es decir, alguien que tenga ADSL puede soportar una conexión de 4KB/s tranquilamente, pero alguien con módem irá justo para poder seguir una de 3KB/s. Así que tendremos que decidir este aspecto con el bitrate y la frecuencia.

El bitrate define la calidad de compresión, es decir, en cuantos bits definirá cada nota; cuanto más alto sea más tonos podrá emitir; en cambio, el bit rate definirá cuantas veces por segundo se emite esta señal.

Cuanto más alta sea la frecuencia mejor será la nitidez del sonido (mejores agudos), pero aumenta la cantidad de información a enviar, así que definiremos siempre la máxima frecuencia que nos permita el bitrate escogido. En cambio el bitrate define la calidad, que se nota mucho con las comunicaciones por satélite, gsm... que utilizan compresión MPEG, las cuales parecen que hablen a base de burbujas y tengan agua de por medio.

Aquí he hecho un cuadro sobre las calidades del mp3:

- ▶ Bitrate (kbps) freq (Hz) KB/s
- ▶ 8 22.050 1 Mala calidad
- ▶ 16 16.000 2
- ▶ 16 22.050 2 Aceptable
- ▶ 24 16.000 3
- ▶ 24 22.050 3 Razonablemente buena
- ▶ 32 22.050 4
- ▶ 32 44.100 4 Muy buena calidad

Clientes

Por Internet, encontraremos a montones...

Radio en Internet: ¿Cómo hacer streaming?

*nix: – mode text: ogg123, mpg123

mode graphic: xmms, linamp

win12 – winamp, wmp

mac – macamp

Por [Pancake/CatHack!](#)

Puedes leer la versión en catalá de este documento en la siguiente url: <http://pancake.host.sk/?t=stream>



Esta publicación esta bajo la licencia [creative commons](#), ello no evita la publicación de otros materiales en otro tipo de licencias

libres. Por tanto, se permite difundir, citar y copiar literalmente sus materiales, de forma íntegra o parcial, por cualquier medio y para cualquier propósito, siempre que se mantenga esta nota y se cite procedencia. Suburbia no asume ninguna responsabilidad por los comentarios y artículos que envían los participantes en este sitio. Toda la responsabilidad para verificar la veracidad y los derechos de reproducción de un envío corresponden al autor/a que lo publica. Al publicar material en este sitio, el o la autora del envío asume que puede ser redistribuido libremente.