

1 Instalacion de Servidor de Audio en Debian

1.1 Sumario

- 1 Introducción a los servicios de audio
- 2 Medios de transmisión
- 3 Componentes que intervienen en el proceso
- 4 Modos de proporcionar el audio
- 5 Frecuencia o tasa de muestreo
- 6 Formatos de Audio
- 7 Streaming
 - ◆ 7.1 Formas de Difusión
- 8 Servidores de Audio en Debian
 - ◆ 8.1 Shoutcast
 - ◇ 8.1.1 Servidor de streaming de audio MP3 con Shoutcast en Debian
 - ◆ 8.2 Icecast
 - ◇ 8.2.1 Instalación de Icecast en Debian
 - ◇ 8.2.2 Instalación del servidor, cliente y herramientas
 - ◇ 8.2.3 Configuración del programa codec ices2
 - ◇ 8.2.4 Como conectarnos a la emisora
 - ◇ 8.2.5 Arranque automático de la Emisora Icecast2 y el cliente Ices2
 - ◇ 8.2.6 Configuración de ICECAST2 para reproducir ficheros MP3
 - ◇ 8.2.7 Direcciones útiles para trabajar con ficheros ogg
- 9 Configuración de drivers de Audio y reproductor MP3 en Debian

1.2 Introducción a los servicios de audio

A lo largo del tiempo hemos visto y conocido las diferentes formas de transmisión de audio, por ejemplo la radio, que permite cubrir grandes distancias en su difusión, o mediante reproductores de cinta (a punto de extinguirse), Cd's, DVD, etc. Gracias al desarrollo tecnológico y a la aparición de Internet, se ha generado una alta demanda de servicios que proporcionen audio a través de la Red.

La posibilidad de escuchar emisoras de radio de cualquier país como si estuvieran al lado de casa, es una opción muy interesante, y muchas de las clásicas emisoras de radio se han modernizado para facilitar este tipo de servicios a sus oyentes.



El servicio de audio surge para proporcionar una alternativa a la transmisión tradicional, adaptándose a las nuevas necesidades de los usuarios, y para utilizarse sobre todo en Internet.

1.3 Medios de transmisión

Existen diferentes maneras para conseguir transmitir un vídeo a través de la red a muchos usuarios a la vez. Una de las opciones más razonables y la única cuando queremos emitir un evento en directo es el **streaming**. Típicamente, para poder ver un vídeo en el ordenador se necesita que el archivo se encuentre en el equipo. El vídeo, sin embargo, es un contenido que requiere archivos muy grandes, por lo que la descarga del mismo desde la red puede hacerse tediosa, e inviable cuando se precisa de la visión en tiempo real. Con el streaming este problema se soluciona: el usuario **cliente** simplemente hace clic sobre un link en una página Web y el vídeo comienza a reproducirse desde otro ordenador **servidor** - sin necesidad de que éste sea el del usuario.

Un ejemplo de este tipo de transmisión sería la página de vídeo bajo demanda <http://www.youtube.com>: cuando vemos un vídeo a través de esta popular página éste no se encuentra en nuestro ordenador, sino que se recibe en tiempo real desde un servidor remoto. En este caso el vídeo está previamente guardado, pero permite aclarar los conceptos que nos llevarán al streaming como sistema de emisión de eventos en tiempo real.

1.4 Componentes que intervienen en el proceso

En primer lugar se deben conocer qué elementos son necesarios para realizar este proceso y qué opciones de transmisión existen en la actualidad. Dos de estos elementos han sido introducidos ya anteriormente: el cliente y el servidor, es decir, el usuario que quiere ver el vídeo y la máquina que los posee y que controla el acceso de los usuarios. Además, se precisa de una cámara que registre el evento y de un equipo con una tarjeta capturadora que codifique el vídeo y el audio (en nuestro caso) y lo mande al servidor. Típicamente, el equipo que codifica el audio y vídeo y el servidor es el mismo, aunque es posible hacerlo por separado. Si la transmisión, en lugar de hacerse en directo, se realiza a través de un archivo pre-grabado el sistema se denomina Vídeo bajo Demanda (VoD).

1.5 Modos de proporcionar el audio

El servidor de audio puede proporcionar el audio a nuestros oyentes de diferentes formas:

- **Descarga directa:** el archivo se aloja en un servidor web o ftp. Para proporcionar estos archivos simplemente se necesita instalar un servidor web HTTP, de forma que atienda las peticiones de los clientes, para la descarga del audio mediante sus navegadores o software adecuado. En este método el usuario tendrá que descargar el archivo previamente antes de poder escuchar su contenido. La forma de conseguir que la reproducción sea más rápida proviene de conseguir una mayor velocidad en la descarga. Una vez descargado el audio se podrá reproducir de forma instantánea las veces que queramos.
- **Descarga progresiva:** permite la reproducción del audio de forma simultánea a la descarga del archivo que lo contiene. Esta forma presenta inconvenientes debido a las diferencias en la velocidad de transmisión y de reproducción, ya que el reproductor se podría quedar en algún momento sin sonido para reproducir.
- **Streaming:** se descarga tan solo la información necesaria a reproducir, permitiendo realizar ajustes en la calidad de la reproducción y ancho de banda utilizado para evitar cortes.

1.6 Frecuencia o tasa de muestreo

Para poder transmitir un flujo de sonido a través de una red, debe existir una digitalización previa del mismo, convirtiendo el audio en paquetes antes de enviarlos al destinatario. La digitalización se realiza usando técnicas de muestreo de la señal analógica que genera el emisor del audio y obteniendo una señal digital. La frecuencia o tasa de muestreo indica el intervalo de tiempo transcurrido entre cada una de las muestras tomadas de la señal analógica. Cuanto mayor es el número de muestras o frecuencia, mejor es la calidad de la señal digital resultante.

Hay que destacar también que el oído humano tiene un umbral a través del cuál el aumento de calidad es imperceptible. Véase: [Umbral de audición en la Wikipedia](#)

Los muestreos se cuantifican con un determinado número de bits; cuanto mayor es el número de bits utilizados mejor calidad.

A veces se utilizan técnicas de compresión, que permiten reducir el tamaño de los archivos, con lo que se mejora la descarga de los mismos, pero tienen una contrapartida y que es la pérdida de calidad.

El software que se encarga de realizar la digitalización y compresión del audio a un formato digital es lo que se conoce como **codificador/decodificador** o técnicamente **codec**.

La secuencia de bits resultante de la codificación se conoce como **bitrate** (tasa de bits). Este bitrate puede ser constante en todo el archivo **CBR** (Constant Bit Rate) o **VBR** (Variable Bit Rate). El VBR usa más o menos bits en la codificación dependiendo del tipo de información transmitida. Por ejemplo para transmitir un silencio o sonidos monótonos, mientras que el CBR utiliza un número de bits constante durante todo el fichero.

Cuando se transmite un flujo de audio se pueden usar varios canales; es decir podremos enviar una o varias señales de audio en el mismo flujo de información. Todos sabemos lo que es el **Mono** o el **Estéreo**. A éstos típicos flujos de audio, podremos añadir también las opciones **multicanal** (5.1 - que transporta hasta 6 canales de audio, 6.1 etc..)

1.7 Formatos de Audio

Podemos destacar dos categorías principales en los formatos de audio:

- **Con pérdida:** comprimen el audio, eliminando la información que no es perceptible por el oído humano. Es decir aquellas frecuencias que no están en el rango de percepción serán eliminadas.
- **Sin pérdida:** también realizan compresión, pero no eliminando frecuencias, sino eliminando redundancias en la señal de audio.

Los formatos de audio más utilizados son:

- **MP3:** El formato de audio más conocido, ya que fue de los primeros en aparecer y supuso un cambio en el uso de los dispositivos de audio. Cuando apareció dicho formato nos quedamos maravillados con la posibilidad de tener en un CD cientos de canciones MP3, cuando antes sólo podíamos escuchar de 12 a 15 canciones de un único grupo de música. Es un formato con pérdida, es sólo estéreo y puede utilizar un bitrate CBR o VBR. El máximo de bitrate que se puede obtener es 320 kbit/s. Por ejemplo la calidad típica de radio FM es 96 kbit/s, el audio CD es 192 kbit/s y para la voz (que pueda ser reconocida) 4 kbit/s. Es un formato propietario y requiere el pago de una licencia. La extensión de los archivos es .mp3 y su tipo MIME es "audio/mpeg" o "audio/MPA". *(El tipo MIME de un archivo es simplemente una descripción de lo que es el archivo. Normalmente se usan en el lado del servidor para poder decir, por ejemplo, que los PDF son de tipo application/pdf, o que las imágenes gif son de tipo image/gif; esto sirve para que el browser escoja la manera apropiada de mostrar cada elemento. Sólo con la extensión (.pdf o .gif) no bastaría).*
- **OGG VORBIS:** Es un formato libre de código abierto desarrollado con licencia GNU GPL, por lo que no es necesario el pago de licencias. Ofrece mucha compresión y con poca pérdida de calidad. Ogg es el contenedor de audio, video y metadatos del fichero y Vorbis es el esquema de compresión utilizado. En este tipo de archivos se puede almacenar información extra sobre autor, álbum, etc, aunque también esto se puede realizar en el fichero mp3. La extensión utilizada es .ogv, .oga, .ogx y .ogg y su tipo MIME es "application/ogg".
- **WAV:** Desarrollado por Microsoft, se utiliza para la grabación de archivos compactos o CD, a una frecuencia de 44 Khz y cuantificación de 16 bits. No usa compresión, así que no es muy recomendable para su uso en Internet. Su extensión es .wav
- **WMA:** Windows Media Audio, también desarrollado por Microsoft, realiza compresión de archivo y consigue adaptarse a la velocidad de conexión. Su extensión es .wma
- **Real Audio:** Lo desarrolló Real Networks, y se utilizó mucho en Internet en su tiempo, ya que fue de los primeros en emplearse en transmisiones de tiempo real. Necesita de un reproductor específico Real Player. La extensión de los archivos es .ram, .rm o .rmm
- **FLAC:** Son las siglas de Free Lossless Audio Codec. Es un formato desarrollado dentro del proyecto Ogg, para codificar audio sin pérdida de calidad. La extensión de los archivos en este formato es .flac o .fla

[Más información sobre formatos de audio en Wikipedia](#)

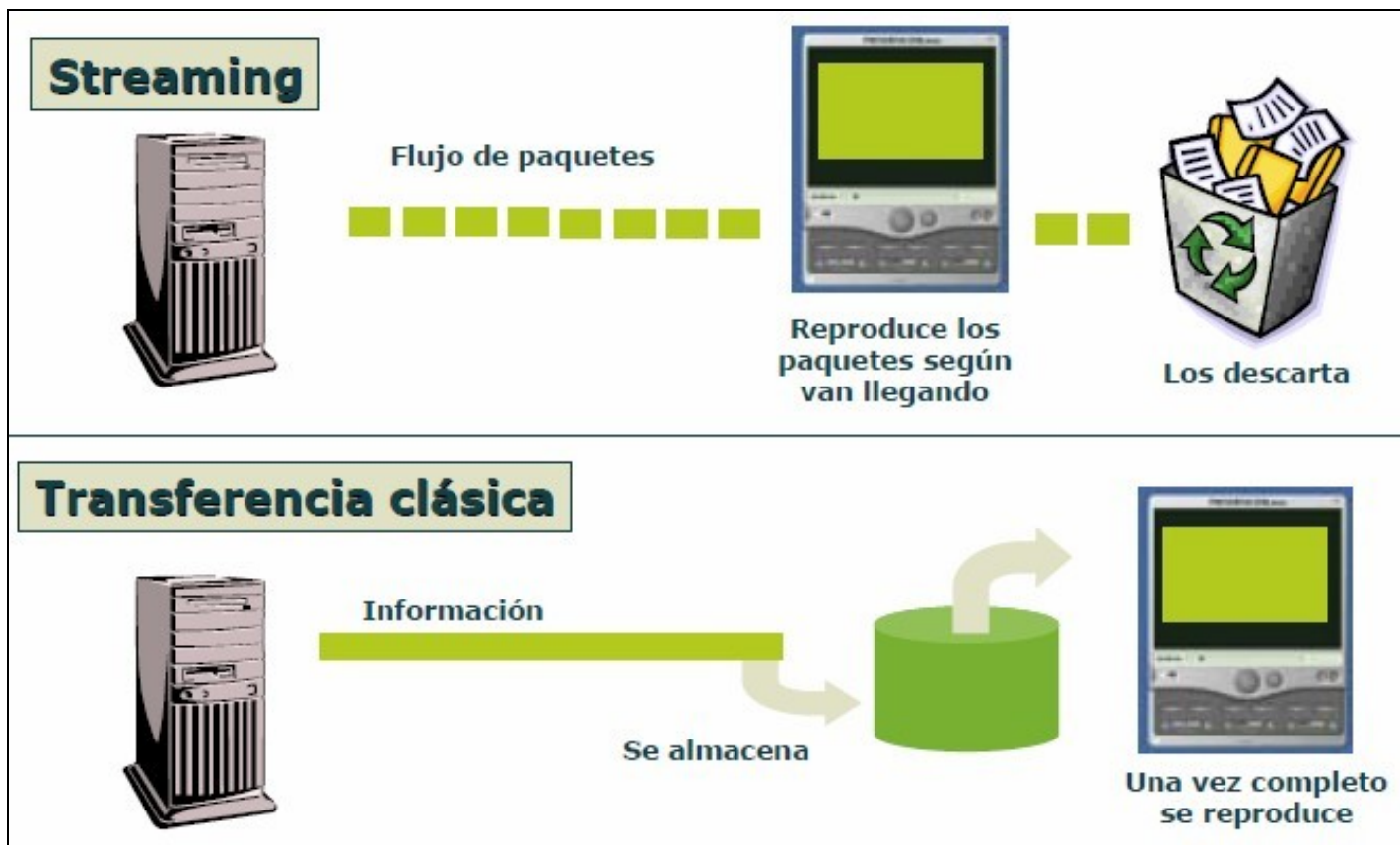
1.8 Streaming

Ya hemos mencionado anteriormente lo que es el streaming, que permite ajustar la calidad de los contenidos a transmitir, y por tanto el ancho de banda utilizado. No realiza descargas completas de los contenidos, sino descargas temporales de lo que se está reproduciendo en un determinado momento. El objetivo principal es conseguir reproducciones sin cortes y con la calidad deseada.

... El streaming se diferencia de una simple transferencia de ficheros en que el cliente reproduce la información mientras la está recibiendo en vez de esperar a recibirla completa antes de reproducirla. De hecho, un cliente de streaming podría no descargar el vídeo/audio, simplemente reproducir la información de cada uno de los paquetes e ir descartándolos. ? (extraído de Apple Quicktime)



En la imagen inferior, vemos una comparativa de lo que es Streaming y la transferencia clásica de archivos.



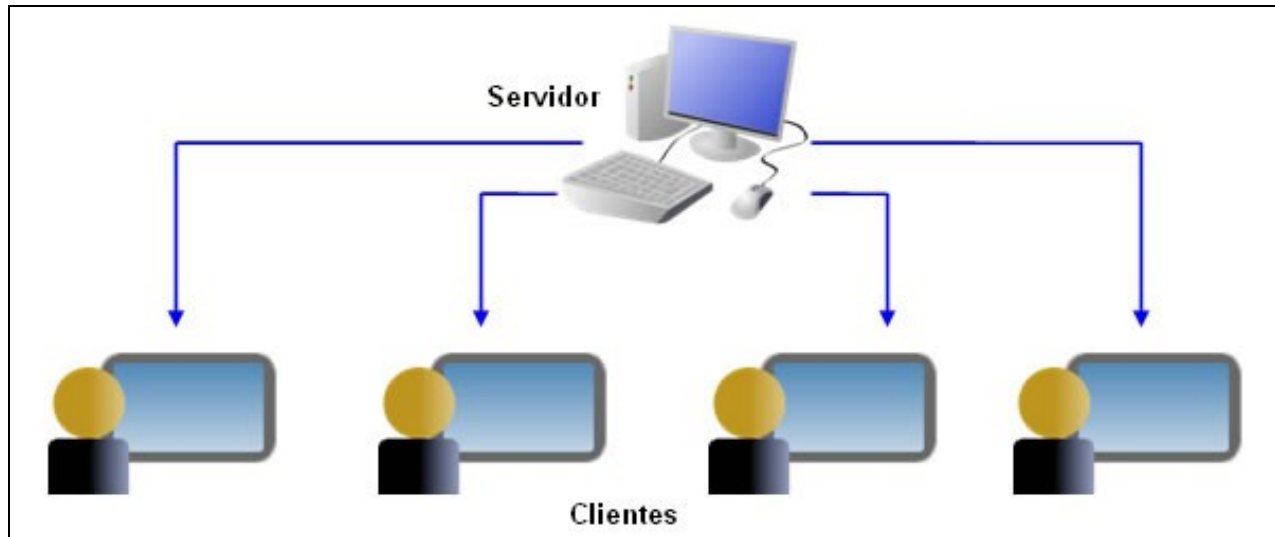
1.8.1 Formas de Difusión

La forma en la que se envía la información a diferentes usuarios permite hacer una clasificación de las posibilidades que ofrece el servicio.

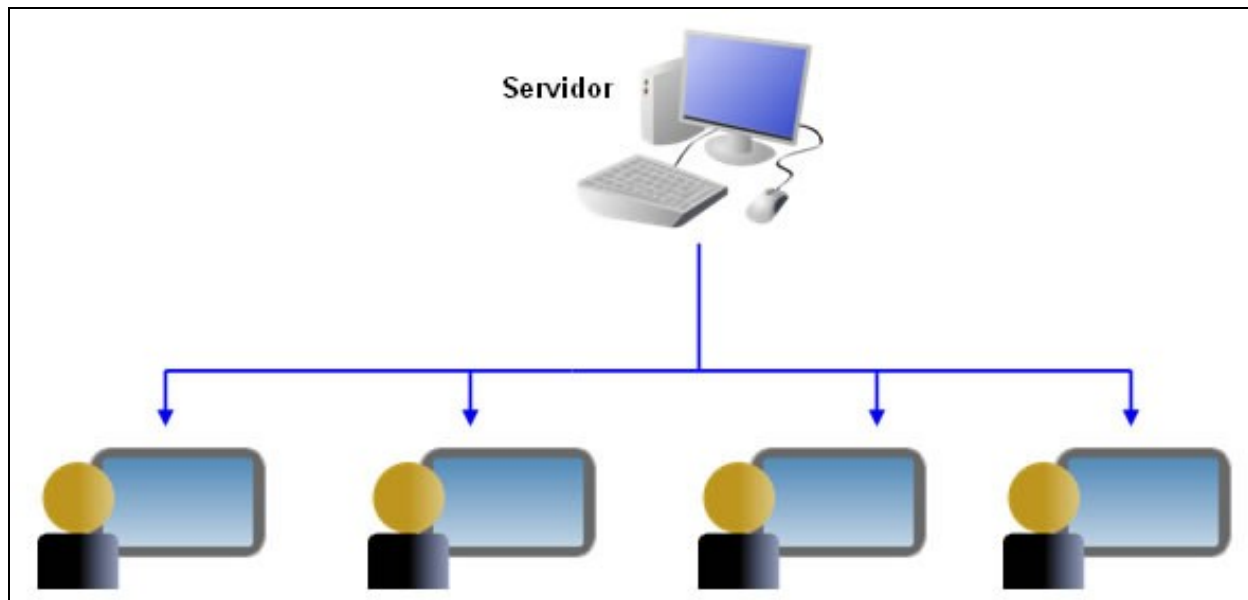
- Si cada usuario que accede al audio obtiene un flujo de datos independientemente, el proceso se denomina **Unicast**.
- Si, por el contrario, el mismo flujo de datos se utiliza para servir el audio o el video a varios usuarios se dice que se está haciendo **Multicast**. Se suele usar en streaming en tiempo real y se usan routers como dispositivos de interconexión, configurados correctamente para transmitir dichos flujos de información. Por ejemplo [Imagenio de Movistar](#) usa este tipo de transmisión.
- Cuando el servidor envía el flujo de datos a todos los equipos de la red, quieran o no recibir esa información se denomina **Broadcast**. Ésto genera gran cantidad de información y puede llegar a saturar la red.

Es importante conocer en qué condiciones se quiere emitir el contenido multimedia, puesto que definirá el ancho de banda necesario en el servidor para que la transmisión sea correcta.

En un escenario de **Unicast**, el método más simple de realizar streaming, el ancho de banda necesario aumenta linealmente en relación al número de usuarios; a mayor número de usuarios, más ancho de banda consumido.



En **Multicast**, sin embargo, este problema se solventa enviando un único flujo de datos al que se conectan los usuarios, por lo que reduce drásticamente el ancho de banda necesario para la transmisión del contenido.



Al tratarse de una transmisión en tiempo real, hacen falta protocolos ligeros para llevar el control durante la transmisión y evitar en todo lo posible pérdidas de información o demoras en la reproducción.

Algunos de los **protocolos** más utilizados en streaming son:

RTSP, RTP, RTCP y UDP.

- [Información sobre RTP y RTSP en Wikipedia](#)
- [Presentación PDF sobre protocolos multimedia en Internet \(en inglés\)](#)

Alguna información y recursos obtenidos de:

<http://recursostic.educacion.es/observatorio/web/es/component/content/article/19-monograficos/567-monografico-introduccion-al-streaming>

1.9 Servidores de Audio en Debian

Existe una amplia variedad de software para la implementación de servidores de audio, de entre los que podemos destacar **Shoutcast** o **Icecast**.

1.9.1 Shoutcast

En la página de [Shoutcast.com](http://www.shoutcast.com) se muestra un portal con todas las emisoras de radio que están transmitiendo utilizando su tecnología (en Abril de 2013 cuentan con 51.201 emisoras de radio!).

Si queremos montar nuestro propio servidor de streaming usando shoutcast, podremos hacerlo a través de sus guías de instalación y configuración:

- <http://www.shoutcast.com/broadcast-tools>
- http://wiki.winamp.com/wiki/SHOUTcast_Broadcaster

Su software de servidor está disponible tanto en versiones para Windows como para Linux y en distintas plataformas.

- Cuando tenemos instalada la emisora y funcionando, podremos configurarla para que automáticamente publique en la página de <http://www.shoutcast.com> la URL y el fichero e información que estamos reproduciendo, para que otros usuarios nos puedan localizar y escuchar nuestra música.

1.9.1.1 Servidor de streaming de audio MP3 con Shoutcast en Debian

Si queremos aparecer en el directorio público de Shoutcast y transmitir en formato MP3, necesitaremos comprar una licencia del codec MP3 (5 dólares aproximadamente).

Para emitir en formato ogg no es necesaria licencia.

Si queremos hacerlo a nivel privado en nuestra red local no es necesario adquirir la licencia MP3.

```
# Descargamos Shoutcast y lo descomprimos:
mkdir shoutcast
cd shoutcast

wget http://download.nullsoft.com/shoutcast/tools/sc_serv2_linux_x64-latest.tar.gz
tar xvzf sc_serv2_linux_x64-latest.tar.gz

# Configuramos Shoutcast:
nano sc_serv_basic.conf
portbase=8000
password=abc123.

# Arrancamos Shoutcast:
root@ns2:~/shoutcast# ./sc_serv sc_serv_basic.conf

# Descargamos Shoutcast transcoder y lo descomprimos:
mkdir trans
cd trans

wget http://download.nullsoft.com/shoutcast/tools/sc_trans_linux_x64_10_07_2011.tar.gz
tar xfzv sc_trans_linux_x64_10_07_2011.tar.gz

# Configuramos Shoutcast transcoder:
nano sc_trans_basic.conf
serverip_1=127.0.0.1
serverport_1=8000
password_1=abc123.
playlistfile=/etc/ices2/playlist.txt

# Arrancamos Shoutcast transcoder:
./sc_trans sc_trans_basic.conf
```

1.9.2 Icecast

Icecast es otro ejemplo de software gratuito que se puede utilizar para montar un servidor de audio, tanto en Linux como en Windows. Icecast está desarrollado por la fundación <http://www.xiph.org/> que se encarga del proyecto.

- Icecast soporta Ogg Vorbis y MP3.
- Gestiona la autenticación de clientes y proveedores de audio.
- Proporciona información al cliente acerca del estado.
- Puede proporcionar varios streamings de forma simultánea.
- Permite la emisión modo live de audio.
- Se puede variar la calidad de audio y limitar el número de clientes que se pueden conectar.

La página oficial de Icecast es <http://www.icecast.org/> y también disponen de un directorio de emisoras en streaming que están usando su tecnología en: <http://dir.xiph.org/index.php>

Para su instalación y configuración en Windows visitar: http://www.duxpoint.com/tutorial_icecast/icecast.htm

Aquí nos vamos a centrar en la instalación y configuración del servicio bajo Linux Debian.

1.9.2.1 Instalación de Icecast en Debian

Icecast es un servidor de flujos de media (media streaming) que soporta Ogg y Mp3. Con programas tales como Winamp o Xmms, cualquier usuario se puede conectar a tu servidor y escuchar contenidos de audio en tiempo real, o en otras palabras, puedes tener tu radio por internet.

Ahora bien, **Icecast** por si solo no hace mucho, ya que a lo sumo permite entregar archivos ogg o mp3 de forma directa, tal cual como lo haría Apache. Para hacer que funcione como una radio, necesitamos lo que se llama un cliente de streaming, que entregue contenidos a Icecast para que funcione. Haciendo una analogía con la radio, Icecast es la antena emisora, en tanto que el cliente es la consola que tiene conectados los micrófonos y el reproductor de CD.

El cliente "oficial" de Icecast es **ices**. La versión 1 permite usar Mp3, en tanto que la 2, más poderosa, permite utilizar Ogg, y será la que instalaremos por tratarse de un formato libre.

1.9.2.2 Instalación del servidor, cliente y herramientas

- Para **instalar Icecast en Debian** realizaremos los siguientes pasos:

```
# Mediante este comando instalamos el cliente ices2, el servidor icecast2 y las utilidades para
# trabajar con ficheros vorbis.
apt-get install icecast2 ices2 vorbis-tools

# Una vez instalado tendremos que configurar el servidor editando el fichero icecast.xml:
nano /etc/icecast2/icecast.xml

# Inicialmente modificaremos los apartados dedicados al usuario admin y a su contraseña,
# y también las contraseñas necesarias para poder transmitir al servidor.
# Eso se modifica en la sección de authentication:
# En source-password pondremos una clave para poder transmitir hacia el icecast con un cliente.
# En relay-password se pone una clave para permitir hacer reenvío a otros servidores.
# Y por último se define el usuario y contraseña de administración.

<authentication>

    <source-password>transmisor</source-password>

    <relay-password>reenviador</relay-password>

    <admin-user>admin</admin-user>
    <admin-password>hola</admin-password>
</authentication>

# Seguimos con el mismo fichero y configuramos la sección del hostname
# Es la dirección que usarán los clientes para conectarse a nuestro servidor.

<hostname>www.pruebas.local</hostname>
```

- Cuando hemos realizado las modificaciones en el fichero icecast2.xml, ya podemos **habilitar el arranque de Icecast**.

```
# Editamos el fichero /etc/default/icecast2
nano /etc/default/icecast2

# En la última línea de ENABLE la ponemos a true
# Edit /etc/icecast2/icecast.xml and change at least the passwords.
# Change this to true when done to enable the init.d script
ENABLE=true
```

- Para arrancar el servicio lo hacemos con:

```
service icecast2 start
```

```
# Y obtendremos algún mensaje como:
Starting icecast2: Starting icecast2
Detaching from the console
icecast2.
```

```
# Otros comandos posibles de gestión de Icecast son:
```

```
service icecast2 force-reload reload restart start stop
```

- Para administrar via web Icecast, si no hemos hecho modificaciones especiales en los puertos, en la sección de <listen-socket> del fichero /etc/icecast2/icecast.xml, por defecto escuchará en el puerto 8000. Nos conectaremos con nuestro cliente web a la dirección del servidor de Icecast2, puerto 8000 y veremos algo como la siguiente captura:



Pulsamos en administración y accederemos con nuestro usuario y contraseña de administración de Icecast:

The image shows a web browser window displaying the 'Icecast2 Status' page. The browser's address bar shows the URL '192.168.1.25:8000'. The page has a dark background with the 'Icecast2 Status' logo and title in white. Below the title, there is a navigation bar with three links: 'Administration', 'Server Status', and 'Version'. At the bottom of the page, there is a text link: 'Support Icecast development at www.icecast.org'. A modal dialog box titled 'Identificación requerida' is overlaid on the page. The dialog contains a question mark icon, a message stating 'http://192.168.1.25:8000 está solicitando un nombre de usuario y una contraseña. El sitio dice: "Icecast2 Server"', and two input fields labeled 'Nombre de usuario:' and 'Contraseña:'. At the bottom of the dialog are two buttons: 'Aceptar' and 'Cancelar'.

Y una vez dentro veremos la página de Gestión de Icecast:



Icecast2 Admin

[Admin Home](#) [List Mountpoints](#) [Move Listeners](#) [Index](#)

Global Server Stats

admin	icemaster@localhost
client_connections	20
clients	1
connections	21
file_connections	12
host	192.168.1.25
listener_connections	0
listeners	0
location	Earth
server_id	Icecast 2.3.2
server_start	Tue, 24 Apr 2012 11:43:44 +0200
source_client_connections	0
source_relay_connections	0
source_total_connections	0
sources	0
stats	0
stats_connections	0

1.9.2.3 Configuración del programa codec ices2

Una vez tenemos instalado el servidor, ahora tendremos que configurar el codificador cliente, es decir quién va a poner la música, para que el servidor la distribuya a todos los oyentes.

```
# Nos crearemos un directorio para el codificador ices2 y copiaremos el fichero de ejemplo
# de configuración que viene incluido en ices2:
mkdir /etc/ices2
cp /usr/share/doc/ices2/examples/ices-playlist.xml /etc/ices2/

# Crearemos la ruta para los logs de ices2 (según la configuración por defecto del xml) con:
mkdir /var/log/ices2

# El archivo de logs se creará en esa carpeta, pero tenemos que asegurarnos de configurarlo
# correctamente en el fichero /etc/ices2/ices-playlist.xml
nano /etc/ices2/ices-playlist.xml

<logpath>/var/log/ices2</logpath>
<logfile>ices2.log</logfile>
```

```

# Una vez copiado editaremos el fichero de configuración del codificador y lo adaptamos a los
# parámetros que tengamos en nuestro servidor icecast:
nano /etc/ices2/ices-playlist.xml

# En la sección de <stream> configuraremos los datos que se mostrarán en la portada de la emisora:

<stream>

    <metadata>
        <name>San Clemente Radio Station</name>
        <genre>Pop, Folk</genre>
        <description>Emisora de música Pop y Noticias de Santiago</description>
    </metadata>

# En la sección de <input> se modifican los datos de la lista que contiene la música que queremos
# reproducir en nuestra emisora. Aquí configuramos el fichero con la lista de canciones a reproducir.
# Indicaremos la ruta dónde se almacena ese fichero, en este caso lo almacenamos en la misma carpeta
# dónde se encuentra ices2 /etc/ices2/musica.txt
nano /etc/ices2/ices-playlist.xml

    <input>
        <module>playlist</module>
        <param name="type">basic</param>
        <param name="file">/etc/ices2/musica.txt</param>

        <param name="random">0</param>

        <param name="restart-after-reread">0</param>

        <param name="once">0</param>
    </input>

# Por último adaptaremos los datos de la conexión, para que nuestro codec se pueda conectar al servidor
# de icecast correctamente y enviarle la música:
<instance>

    <hostname>www.pruebas.local</hostname>
    <port>8000</port>
    <password>transmisor</password>
    <mount>/endirecto</mount>

# Hay otras secciones en el fichero en las cuales podemos modificar el bitrate de nuestro codec.
# Aquí modificaremos la calidad a la cuál vamos a transmitir los ficheros a nuestro servidor icecast.
# Es importante este parámetros para hacer cálculos de ancho de banda según las posibilidades
# de nuestro servidor:
    <encode>
        <nominal-bitrate>192000</nominal-bitrate>
        <samplerate>44100</samplerate>
        <channels>2</channels>
    </encode>

```

Ahora nos faltaría simplemente generar el archivo /etc/icecast2/musica.txt que definimos en el fichero de configuración de ice2. Ese archivo contendrá la lista de música que queremos que se reproduzca en la emisora.

Una forma rápida de generar ese listado consiste en buscar los archivos ogg o mp3 que tengamos en nuestro servidor y redireccionar la ruta de esos archivos añadiéndolos al fichero musica.txt:

```
find / -iname "*.ogg" >> /etc/ices2/musica.txt
```

Para poner a funcionar nuestro codec y que comience a transmitir música al servidor se hace con:

```

# Indicamos a ices2 dónde está el fichero de configuración de nuestro CODEC,
# y arrancamos el servicio en background:
ices2 /etc/ices2/ices-playlist.xml &

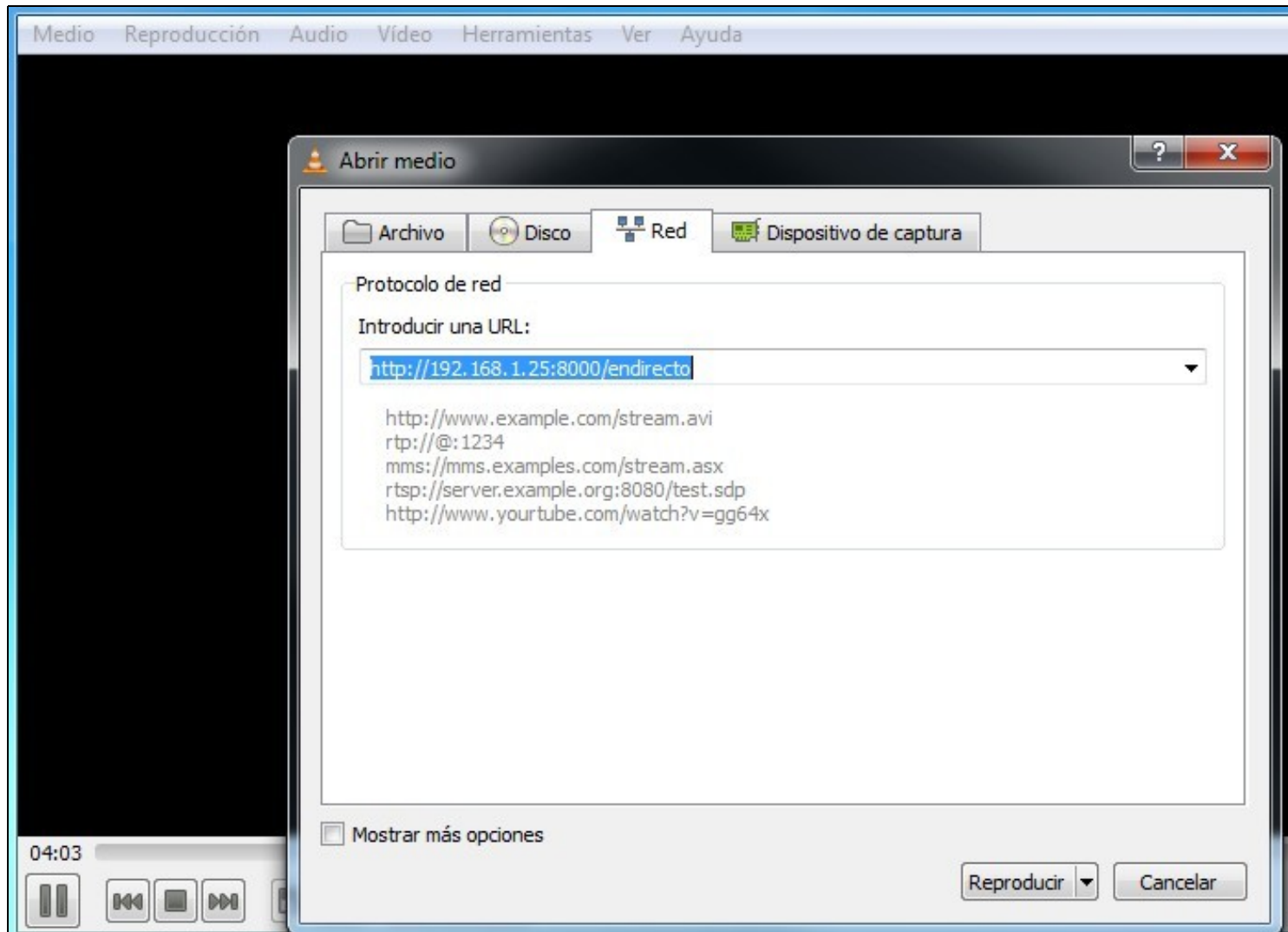
```

```
# Si deseamos parar el codec ices2 lo podemos hacer con:  
kill -9 `pidof ices2`
```

1.9.2.4 Como conectarnos a la emisora

En principio para poder conectarnos a la emisora icecast2 tendremos que usar algún reproductor que soporte ogg. Por ejemplo [Videolan](#), lo soporta.

Tendremos que abrir Videolan, y en el **Menú Medio** => **Abrir Volcado de Red CTRL+N**:



Introduciremos la URL de la emisora (**http://IP:Puerto/puntoMontaje**) y pulsaremos Reproducir:

```
# La URL de la emisora puede ser algo del estilo:  
http://192.168.1.25:8000/endpoint
```

Podríamos añadir nuevos ficheros a la lista de reproducción sin parar la emisora, simplemente editando el fichero musica.txt que contiene los ficheros, o bien añadiendo los nuevos tal y como hicimos antes.

Si tenemos la precaución de meter todos los ficheros de música dentro de una misma carpeta podremos ejecutar siempre la misma orden:

```
# Por ejemplo para añadir todos los ficheros de /root/micoleccion a la emisora:  
find /root/micoleccion/ -iname *.ogg > /etc/ices2/musica.txt
```

1.9.2.5 Arranque automático de la Emisora Icecast2 y el cliente Ices2

Ices2 no viene con un script de inicio que nos permita arrancarlo o pararlo como hacemos con Icecast2. Icecast2 se arranca automáticamente en el boot del sistema, pero Ices2 no, por lo que es un poco molesto cada vez que arrancamos el sistema el tener que arrancar el Ices2.

Sin embargo podemos modificar el script de inicio de Icecast2 para que añada secciones para Ices2. Aquí se incluye un ejemplo:

```
nano /etc/init.d/icecast2

#!/bin/sh
#
# icecast2
#
#           Written by Miquel van Smoorenburg <miquels@cistron.nl>.
#           Modified for Debian
#           by Ian Murdock <imurdock@gnu.ai.mit.edu>.
#
#           Further modified by Keegan Quinn <ice@thebasement.org>
#           for use with Icecast 2
#

PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
DAEMON=/usr/bin/icecast2
NAME=icecast2
DESC=icecast2
ICES=/usr/bin/ices2
ICES_CONFIGFILE=/etc/ices2/ices-playlist.xml

test -x $DAEMON || exit 0

# Defaults
CONFIGFILE="/etc/icecast2/icecast.xml"
CONFIGDEFAULTFILE="/etc/default/icecast2"
USERID=icecast2
GROUPID=icecast
ENABLE="false"

# Reads config file (will override defaults above)
[ -r "$CONFIGDEFAULTFILE" ] && . $CONFIGDEFAULTFILE

if [ "$ENABLE" != "true" ]; then
    echo "$NAME daemon disabled - read $CONFIGDEFAULTFILE."
    exit 0
fi

set -e

case "$1" in
    start)
        echo -n "Starting $DESC: "
        start-stop-daemon --start --quiet --chuid $USERID:$GROUPID \
            --exec $DAEMON -- -b -c $CONFIGFILE
        sleep 3
        start-stop-daemon --start --quiet --exec $ICES $ICES_CONFIGFILE &
        echo "$NAME."
        ;;
    stop)
        echo -n "Stopping $DESC: "
        start-stop-daemon --stop --oknodo --quiet --exec $ICES

        start-stop-daemon --stop --oknodo --quiet --exec $DAEMON
        echo "$NAME."
        ;;
    reload|force-reload)
        echo "Reloading $DESC configuration files."
        start-stop-daemon --stop --oknodo --quiet --exec $ICES
        start-stop-daemon --stop --signal 1 --quiet --exec $DAEMON
        sleep 3
        start-stop-daemon --start --quiet --exec $ICES $ICES_CONFIGFILE
        ;;
    restart)
        echo -n "Restarting $DESC: "
        start-stop-daemon --stop --oknodo --quiet --exec $ICES

        start-stop-daemon --stop --oknodo --quiet --exec $DAEMON
        sleep 3
        start-stop-daemon --start --quiet --chuid $USERID:$GROUPID \
            --exec $DAEMON -- -b -c $CONFIGFILE
```

```

        sleep 3
        start-stop-daemon --start --quiet --exec $ICES $ICES_CONFIGFILE &
        echo "$NAME."
        ;;
*)
        echo "Usage: $0 {start|stop|restart|reload|force-reload}" >&2
        exit 1
        ;;
esac

exit 0

```

A partir de ahora cuando se arranque nuestro servidor también se arrancará el servidor de audio Icecast2 y el codec Ices2.

IMPORTANTE: A partir de ahora, si cuando arranca el Icecast2 se queda en Detaching from console y no os deja hacer nada, eso es debido a que os falta añadir el símbolo & (en Start y Restart) para la línea de \$ICES.

Si ésto os está ocurriendo al arrancar la máquina y no os deja continuar, la manera de solucionarlo es la siguiente:

```

# Reiniciamos la máquina desde Virtualbox.

# Cuando aparece el menú de boot mostrando las opciones de sistemas operativos,
# os poneis en Recovery Mode.

# Entráis como root y editais de nuevo el fichero:
nano /etc/init.d/icecast2

# Dentro de las secciones de start y restart ponéis el símbolo & al final de la línea que
# hace referencia a: ICES:
ICES $ICES_CONFIGFILE &

# Otra forma de acceder en modo Single User (si no os aparece la opción de Recovery Mode):

# Situarse encima del Sistema Operativo.
# Pulsar la letra E
# Buscar la línea que hace referencia a /linux/vmlinuz... y al final de esa línea en:
# .... ro quiet

# Añadimos la letra S (de single user):
.... ro quiet S

# Pulsamos Ctrl+X para Iniciar de nuevo, y seguimos los pasos para solucionar lo del ICES.

```

1.9.2.6 Configuración de ICECAST2 para reproducir ficheros MP3

Podremos tener en ICECAST2 dos puntos de montaje que escuchen en el mismo puerto y que cada uno transmita en un formato de archivo distinto (ogg y mp3).

Para ello necesitamos el cliente ices pero una versión antigua.

```

# Primeramente nos descargamos ices 0.4
wget http://downloads.us.xiph.org/releases/ices/ices-0.4.tar.gz

# Librerías necesarias para compilar:
apt-get install gcc
apt-get install g++
apt-get install libshout3-dev
apt-get install make
apt-get install python-dev
apt-get install libxml2-dev
apt-get install libperl-dev

# Descomprimos el ices-0.4.tar.gz:
tar xvf ices-0.4.tar.gz

#Lo compilamos e instalamos con las siguientes instrucciones:
cd ices-0.4
./configure

```

```

make
make install

# Ruta del archivo de configuración:
/usr/local/etc/ices.conf.dist

# Ejemplo de modificaciones a realizar o añadir al fichero ices.conf.dist
<File>/etc/ices/playlistmp3.txt</File>
<Hostname>dirección-IP</Hostname>
<Port>8000</Port>
<Mountpoint>/mp3</Mountpoint>

# Ejecutamos el cliente
# Si indicamos el fichero de playlist en la configuración usaremos:
ices -c <path_to_config_file> &

# En otro caso podremos ejecutar el cliente indicándole que fichero playlist usar:
ices -c <path_to_config_file> -F <path_to_playlist> &

# Esta sección ha sido documentada gracias a la aportación del alumno:
# Francisco José Carreira Sixto (curso 2012-2013) del IES San Clemente.

```

1.9.2.7 Direcciones útiles para trabajar con ficheros ogg

- Si queremos convertir ficheros **wav a formato ogg**, lo podremos hacer desde la línea de comandos en Debian, con la utilidad **oggenc**
- Para convertir archivos de audio a diferentes formatos (por ejemplo de mp3 a ogg) se puede hacer online a través de la página web: <http://media.io/>

1.10 Configuración de drivers de Audio y reproductor MP3 en Debian

Para configurar el audio en una máquina Debian que funcione en modo texto:

```

# Instalamos los drivers de sonido
apt-get install alsa

# Recargamos los drivers de sonido.
alsa reload

#muestra dispositivos PCI
lspci -v

# Para instalar un reproductor MP3 en modo comandos.
apt-get install madplay

# Como crear una mini-emisora MP3 con netcat.
# Servidor en puerto 32000:
cat *.mp3 | nc -l -p 32000 -w 10

#Parámetros:
-w : wait 10 segundos
-l : listen
-p : port

#Cliente que se conecta al servidor para escuchar la música MP3:
nc IP_del_servidor 32000 | madplay -

```