

Servidor de Corrents d'Àudio i Vídeu basat en Programari i Formats Lliures

Àngela Abad
*Projectista de la Universitat
Politécnica de Catalunya*

Xavier Giró
*Professor de la Universitat
Politécnica de Catalunya*

juny de 2007

Resum

La distribució de senyals audiovisuals per Internet és una tendència a l'alça a la qual la comunitat de programari lliure aporta solucions. Aquesta comunicació explica l'experiència d'instal·lació d'un servidor IceCast de la Fundació Xiph.org per a la transmissió de corrents d'àudio i vídeo utilitzant la família de formats lliures impulsats per la mateixa organització.

1 Introducció

L'augment d'ample de banda de les xarxes i les seves connexions, combinat amb el procés de digitalització i codificació de continguts audiovisuals, ha obert la porta a la seva distribució a través de xarxes telemàtiques. La distribució i consum de senyals audiovisuals a través de xarxes IP com Internet requereix programaris, protocols i formats de dades adaptats als serveis de comunicacions audiovisuals. Aquest tipus de serveis poden ser dividits en dues grans famílies: la de les transmissions en directe o les d'enregistrament per a una reproducció posterior [1]. Aquesta comunicació aporta solucions pel primer dels casos en una arquitectura de client – servidor i explica una experiència a l'hora d'utilitzar solucions de programari lliure i formats lliures. En concret, es proposa la utilització del VLC del projecte VideoLAN com a font i client d'un servidor de corrents audiovisuals basat en els formats lliures de la fundació Xiph.org.

Aquesta comunicació és un resum del projecte de final de carrera d'Àngela Abad en els estudis d'Enginyeria Tècnica de Telecomunicacions, especialitat en So i Imatge, a l'Escola

Universit ria d'Enginyeria Industrial de Terrassa (EUETIT) de la Universitat Polit cnica de Catalunya (UPC). El treball s'ha realitzat sota la supervisi  del professor Xavier Gir  del Grup de Processament de la Imatge (GPI) del departament de Teoria del Senyal i Comunicacions (TSC) de la mateixa UPC.

2 Formats de dades

Els senyals digitals d' udio i v deo poden ser representats en diversos formats que, des del punt de vista de llic ncies, comparteixen problem tiques similars al programari. De la mateixa manera que un dels requeriments del projecte va ser utilitzar programari lliure, tamb  es va considerar oport  treballar amb esquemes de codificaci  lliures per aconseguir la m xima flexibilitat alhora que s'assolia un cost m nim, ja que d'aquesta manera s'eviten les patents o llic ncies restrictives. La comunitat de programari lliure tamb  aporta soluci ns a nivell de formats, encara que avui en dia tenen una pres ncia minorit ria en el mercat. En aquest projecte ens vam fixar amb els formats proposats per la Fundaci  Xiph.org [2] dins el projecte OGG.

2.1 C decs d' udio i v deo: Vorbis i Theora

En moltes aplicacions audiovisuals els corrents d' udio i v deo adquirides per micr fons i c meres s n codificats i comprimits amb una fam lia de programari anomenat c decs.

En el cas de l' udio es va adoptar el c dec Vorbis desenvolupat per Xiph.org com a part del projecte Ogg. Vorbis  s adequat per a  udio de mitjana a alta qualitat a taxes de transfer ncia fixes i variables, i ofereix una qualitat superior a altres formats propietaris com MP3 o WMA a una mateixa taxa de bits.

Pel que fa al v deo la Fundaci  Xiph.org proposa el Theora, un c dec basat en l'antic format VP3, el codi del qual la companyia On2 Technologies va alliberar sota una llic ncia BSD. La fundaci  Xiph.org l'ha refinat i ampliat donant-li el mateix abast de futur que el c dec d' udio Vorbis.

2.2 Contenedor: OGG

Una vegada definides les t cniques de compressi  i el format de les dades de l' udio i v deo el seg ent pas  s escollir el format d'un nivell superior d'abstracci  que els combini i sincronitzi. Aquests s n els formats contenidors i, a nivell de llic ncies i patents, presenten problem tiques similars als codificadors d' udio i v deo. Tenen l'afegit que no tots els contenidors poden acollir a tots els c decs que hi ha al mercat o a les seves combinacions.

En el nostre cas el contenidor OGG  s la soluci  natural per la combinaci  de Vorbis i Theora, dissenyats per treballar conjuntament sota OGG i evitar aix  problemes de sincronitzaci .

L'OGG també és lliure de patents i obert, i fou dissenyat per a donar un alt grau d'eficiència en la reproducció en temps real i en la compressió de fitxers.

3 Programari

L'arquitectura que es persegueix en aquest projecte és la d'una font d'àudio o vídeo que genera un corrent de dades (àudio i vídeo) que queda disponible a un servidor amb interfície web. Els clients utilitzaran el navegador web per accedir a un llistat dels corrents disponibles al servidor i poder-ne sol·licitar la transmissió. La petició i reproducció del corrent es realitza per una tercera aplicació reproductora que, en el nostre cas, coincideix amb el programari difusor.

Descripció del maquinari amb el que s'ha treballat:

CPU	Pentium IV a 3,2 Ghz
RAM	4GB
Disc Dur	160 GB Serial Ata a 7200 rpm
Tarja Gràfica	PCI express de 256 Mb ATI

3.1 Servidor: IceCast

IceCast és un projecte de programari lliure de la Fundació Xiph.org, primerament dedicat a la difusió d'àudio, que a partir de la versió 2, permet la difusió de vídeo. Part del projecte es un servidor, capaç de difondre continguts com Vorbis, per a àudio, i Theora, per a vídeo, sobre el protocol HTTP. L'IceCast ofereix també una interfície web des d'on es pot gestionar l'estat del servidor i explorar els corrents disponibles.

Requereix un programa extern, com a font del flux de vídeo, com el FreeJ, l'iCes o el VLC. En el cas de voler crear un flux d'àudio hi han molts programes externs disponibles per a diversos sistemes operatius.

3.2 Font, client i reproductor: VLC

VideoLan [3] és un projecte de programari lliure que produeix programari per a la recepció i enviament de vídeo. Alliberat sota la GNU/GPL, funciona en moltes plataformes i disposa d'una interfície en català [4]. Inclou característiques molt importants, que permeten crear corrents multicast, des d'un arxiu, un disc o també des d'un webcam (en aquest cas és necessari utilitzar Linux i haver instal·lat Vídeo 4 Linux). Des de la versió 0.86 dóna suport a la distribució de corrents pel mètode Shout cap a un servidor, com per exemple l'IceCast.

4 Instal·lació i configuració

4.1 El Servidor IceCast

El Sistema Operatiu que s'ha fet servir per a la instal·lació i configuració del programari fou una distribució GNU/Linux, concretament, el Kubuntu 7.04 Feisty Fawn. Aquesta distribució ens va donar molt bon resultat a l'hora de configurar la càmera web.

L'IceCast requereix la versió de desenvolupament d'algunes llibreries externes populars, totes elles disponibles als repositoris de Kubuntu i instal·lables amb un gestor de paquets, en el nostre cas, l'*apt-get*. En concret, abans d'instal·lar els paquets *icecast-server* i *icecast2* cal tenir instal·lats el *libxml2-dev*, *libxslt-dev*, *libalogg-dev* i l'*icecast-server*.

Després d'instal·lar els paquets encara cal fer els següents passos per tenir el servidor en marxa:

- Canvieu els permisos de la carpeta */etc/icecast2* (per omissió), per a que l'usuari pugui editar l'interior. Les *xxxx* s'han de substituir pel nom d'usuari.

```
sudo chown -R xxxx:users /etc/icecast2/
```

- Feu una còpia del fitxer *icecast.xml* en un directori creat per l'usuari. Aquest fitxer és la principal eina de configuració del servidor i conté diferents apartats que cal especificar, segons els nostres requeriments. Es tracta d'un fitxer XML que es pot modificar amb qualsevol editor de text. El mateix paquet d'instal·lació ja incorpora un fitxer per omissió on s'hi explica amb detalls els diferents paràmetres.

```
mkdir /home/.../icecast/  
cp /etc/icecast2/icecast.xml /home/.../icecast.xml  
cd /home/.../
```

- Cal crear dos fitxers buits, per a que IceCast generi els seus registres. Canvieu els permisos de la carpeta dels logs, per que l'usuari tingui accés.

```
sudo chown -R xxxx:users /var/log/icecast2/  
cd /var/log/icecast2/  
touch error.log  
touch access.log
```

- L'últim pas és posar en marxa el servidor IceCast:

```
icecast2 -c /home/.../icecast.xml
```

- Es pot comprovar el funcionament del servidor des d'un navegador web accedint a l'adreça *http://IP_servidor:8000*. Si tot ha funcionament correctament, es visualitza la imatge mostrada per la Figura 1.



Figura 1: Intefície web del servidor IceCast

Per altra banda, és necessari comprovar que la xarxa en la qual hi ha el servidor no hi ha restriccions als ports 8000 i 8001 utilitzats per l'IceCast.

4.2 El VLC com a client i reproductor

L'aplicació més popular del VLC és la de treballar com a client i com a reproductor. Pot treballar tant com una aplicació independent o com un connector del Firefox si es vol integrar dins un lloc web. La distribució per omisió descarregable des de la web del projecte o des dels repositoris més populars incorpora totes les funcionalitats. Cal destacar que el VLC és multiplataforma i que es tracta d'un dels reproductors més potents del moment.

- Instal·lació com a client reproductor

```
sudo apt-get install vlc
```

- Instal·lació amb connector

```
sudo apt-get install vlc vlc-plugin-* mozilla-plugin-vlc
```

4.3 El VLC com a font

Els instal·lables del VLC que es distribueixen normalment no tenen activat el mòdul `access_output_shout` necessari per utilitzar-lo com a font del nostre sistema.

- El primer que cal fer és comprovar si el mòdul està instal·lat al nostre sistema.

```
vlc -l | grep shout
```

- El següent llistat, tot i tenir mòduls shout, no inclou el necessari per fer treballar el VLC com a font.

```
VLC media player 0.8.6 Janus
playlist      New winamp 5.2 shoutcast import
shout         Shoutcast radio listings
shout         Shoutcast TV listings
```

En aquest cas cal instal·lar VLC a partir d'una recompilació del codi font. Per tal de no estar pendent d'actualitzacions de programari, proposem una instal·lació totalment separada i independent de la que ja apareix per omisió en nombroses distribucions GNU/Linux.

- Descarregueu i descomprimiu el codi font de l'última versió. En el nostre cas, vam treballar amb la 0.8.6b.

```
wget http://download.videolan.org/pub/videolan/vlc/0.8.6b/vlc-0.8.6b.tar.gz
mv vlc-0.8.6b.tar.gz /usr/local/src/.
cd /usr/local/src
tar xvfz vlc-0.8.6b.tar.gz
```

- Col·loqueu-vos al directori on l'heu descomprimit

```
cd /usr/local/src/vlc-0.8.6b/
```

- Instal·leu els paquets necessaris per compilar el VLC:

```
apt-get build-dep vlc;
```

- Existeixen múltiples opcions, mòduls i còdecs disponibles a l'hora de compilar VideoLan. Es poden consultar amb la instrucció següent:

```
./configure -help
```

En aquest projecte es van utilitzar els següents mòduls: còdec theora, el modul v4l per accedir a dispositius de captura, el mòdul dvb en el cas que es volgués accedir a una tarja de TV analògica o digital, i mòdul shout, que és el que ens permet servir de client font a IceCast.

- Compileu l'aplicació amb els mòduls descrits.

```
./configure --prefix=/opt/vlc-0.8.6b/ --exec-prefix=/opt/vlc-0.8.6b/ --enable-theora --enable-shout --enable-v4l --enable-dvb
make
sudo make install
```

- Comproveu ara que el mòdul Shoutcast està instal·lat

```
./vlc -l | grep shout
VLC media player 0.8.6a Janus
access_output_shout      IceCAST output
playlist                 New winamp 5.2 shoutcast import
shout                    Shoutcast radio listings
shout                    Shoutcast TV listings
```

- Executeu vlc

```
./vlc
```

- S'obrirà la seva interfície gràfica que mostra la Figura 2.

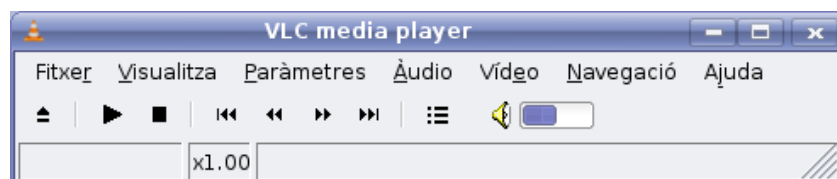


Figura 2: Interfície gràfica del VLC

- Feu clic a 'Paràmetres' -> 'Preferències'. Al menú de l'esquerra seleccioneu 'Corrent de Sortida' -> 'Sortida d'Accés' -> 'Shoutcast'. Aquí es pot configurar el nom del corrent i la seva descripció..

4.4 Instal·lacions complementàries

Hi ha una sèrie d'instal·lacions recomanables per tal de dotar al sistema del màxim de flexibilitat

4.4.1 Els transcodificadors *ffmpeg2theora* i *mencoder*

El paquet *ffmpeg2theora* permet transcodificar fitxers MPEG o AVI a OGG Theora, per la seva banda, el *mencoder* va ser utilitzar per canviar fitxer MP4 a MPEG. Amb un senzill fitxer d'ordres es pot transcodificar un rere l'altre la col·lecció de vídeos que no estiguin codificats amb Theora en origen.

4.4.2 El controlador de la càmera

Per a tenir un bon funcionament de la càmera web, una Logitech QuickCam Express, va ser necessari instal·lar els controladors de la webcam. Per GNU/Linux amb el paquet *spce5xx-source*, mentre que per Windows es va utilitzar el CD-ROM que es proporciona conjuntament amb la càmera.

5 Casos d'ús

5.1 Enviament d'àudio

Aquesta secció explica com fer funcionar el programari que hem instal·lat i configurat a l'apartat anterior per l'enviament d'àudio. Sempre es necessitarà una aplicació font, que en el nostre cas el VLC, però que també podria ser l'Ices de la mateixa Fundació Xiph.org. La font envia el corrent d'àudio cap al servidor, IceCast i aquest el posa a disposició dels clients per a connexions Multicast. Tot i escapar-se de la temàtica d'aquesta comunicació, en el cas de l'àudio també es va comprovar que la tècnica és aplicable al sistema de distribució P2P implementat pel també programari lliure Peercast.

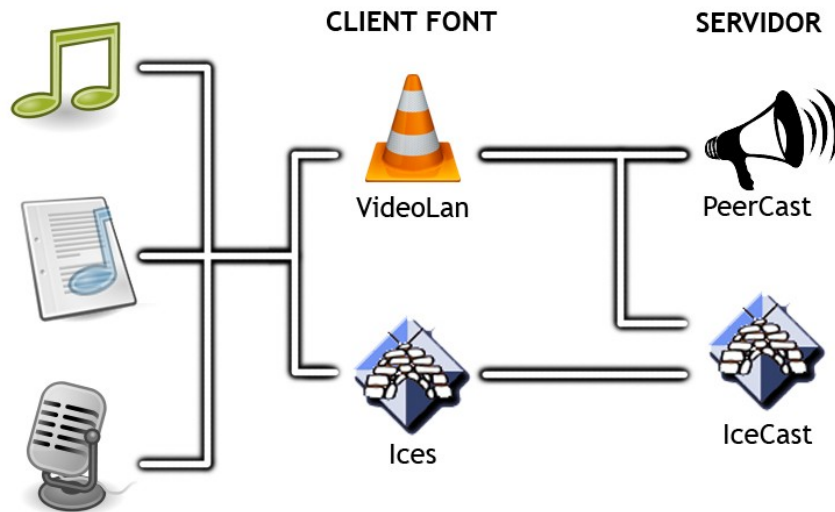


Figura 3: Esquema de corrents d'àudio

El format de l'ordre a utilitzar tindrà aquesta estructura com a base:

```
./vlc -vvv TIPUS_FONT --sout '#standard{access=shout,mux=ogg,url=source:SOURCE-PASSWORD@IP_SERVIDOR:PORT/PUNT_MUNTATGE.ogg}'
```

- El TIPUS_FONT varia depenent del que es vulgui enviar, ja sigui un fitxer d'àudio, una llista de reproducció, o des de l'entrada de la tarja de so, com pot ser el micròfon.
- Amb el paràmetre --sout, li estem indicant que ha d'utilitzar el mòdul shout que hem instal·lat. L'accés també serà del tipus shout.
- L'àudio que s'envii haurà d'estar encapsulat (mux) en un contenidor ogg
- La part de la url, es la més important per poder connectar correctament amb el servidor IceCast.
 - source, es el nom per omisió per a totes les connexions.
 - SOURCE-PASSWORD, és la contrasenya d'accés al servidor per als corrents, que s'ha establert al fitxer de configuració icecast.xml.
 - IP_SERVIDOR, és l'adreça IP a la que hem de dirigir el corrent que enviem.
 - PORT, és el port al que tenim el servidor IceCast, si no s'ha fet cap canvi al fitxer de configuració, el port és el 8000.
 - PUNT_MUNTATGE.ogg, aquest nom l'estableix l'usuari que es disposa a enviar el corrent, per tal de diferenciar les diferents fonts que arriben al servidor IceCast, tal com mostra la Figura 4. Per tant, es pot posar qualsevol nom, però sempre ha d'acabar amb l'extensió ogg.



Figura 4: Exemple de punt de muntatge a la web del servidor Icecast

Aquesta estructura bàsica aporta solucions a diferents casos d'ús.

5.1.1 Fitxer en format OGG Vorbis

En aquest cas només haurem d'utilitzar l'estructura bàsica, degut a que el fitxer ja es troba en el format OGG, no cal codificar-lo, només enviar-lo.

```
./vlc -vvv /home/.../cançò.ogg --sout '#standard{access=shout,mux=ogg,url=source:***@XXX.XXX.XXX.XXX:8000/vlc_music_ogg.ogg}'
```

5.1.2 Fitxer En Format MP3

En aquest cas es pretén enviar un corrent a partir d'un fitxer en format MP3, sense haver-lo convertit prèviament a OGG. En aquest cas és el VLC l'encarregat de codificar-lo, utilitzant el còdec vorbis (acodec=vorbis) i establint la taxa de bits (ab=128). La resta de l'estructura romandrà igual.

```
./vlc -vvv /home/.../cançò.mp3 --sout '#transcode{acodec=vorbis,ab=128}:standard{access=shout,mux=ogg,url=source:***@XXX.XXX.XXX.XXX:8000/vlc_music_mp3.ogg}'
```

5.1.3 Llista de Reproducció

En aquest cas, l'estructura de l'ordre depèn del que hi hagi a la llista de reproducció emmagatzemada en un fitxer d'extensió *.m3u*. Si son fitxers d'àudio en OGG s'utilitza l'estructura bàsica, però si els fitxers son MP3 caldrà transcodificar-los, anàlogament al que es feia amb un únic fitxer. El mateix VLC aporta una interfície per a crear llistes de reproducció d'una manera còmoda, tal com mostra la Figura 5.

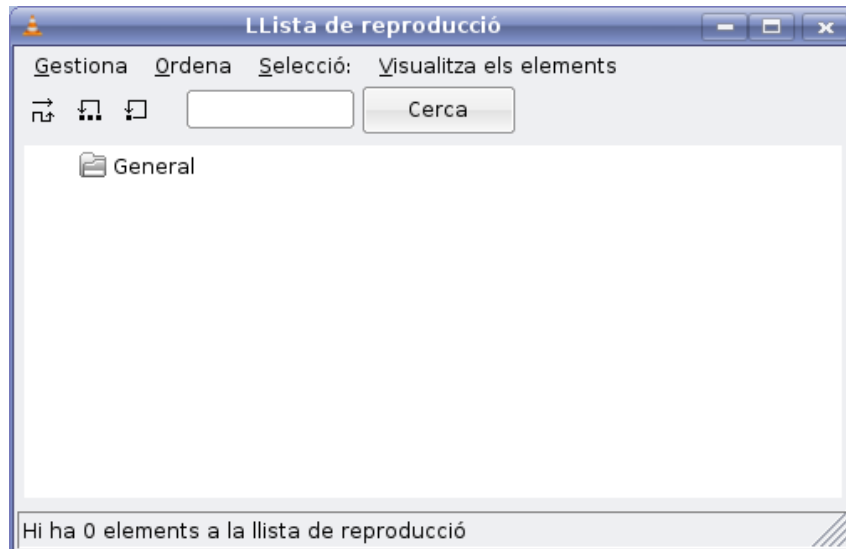



Figura 5: Llista de reproducció del VLC

- Inserir els fitxers que formaran part de la llista.
- Per desar la llista, aneu a: Gestiona>Desa la llista de reproducció
- Un cop creada i desada la llista, tanqueu aquesta interfície gràfica del VLC i continueu executant les ordres des del terminal.
- En el cas que la llista contengui només fitxers OGG, s'utilitza l'estructura bàsica, però no s'ha d'incloure l'extensió al fitxer de llista.

```
./vlc -vvv /home/.../llista_ogg --sout '#standard{access=shout,mux=ogg, url=source:***@XXX.XXX.XXX.XXX:8000/vlc_llista_ogg.ogg}'
```

- En el cas d'utilitzar fitxers MP3, cal codificar-los a OGG abans d'enviar-los.

```
./vlc -vvv /home/.../llista_mp3 --sout '#transcode{acodec=vorbis,ab=128}:standard{access=shout,mux=ogg,url=source:***@XXX.XXX.XXX.XXX:8000/vlc_llista_mp3.ogg}'
```

- Un cop ja s'està enviant el corrent, podem fer que aquesta llista no s'aturi mai, i es reproduïxi en forma de bucle, prement aquest botó: 

5.1.4 Entrada de la Tarjeta de So

Si voleu que el corrent sigui l'entrada de la tarjeta de so, per exemple un micròfon o la sortida d'un receptor de ràdio, s'utilitza el mòdul video4Linux (v4l). Aquesta configuració pot interessar a emissores de ràdio analògica que vulguin distribuir els seus continguts a través d'Internet.

```
./vlc -vvv v4l:/dev/dsp --sout '#transcode{acodec=vorbis,ab=128}:standard{access=shout,mux=ogg,url=source:***@XXX.XXX.XXX.XXX:8000/vlc_micro.ogg}'
```

5.2 Enviament de vídeo

Seguint el mateix esquema que amb l'àudio, ens hem decantat pel VLC com a font i reproductor. La Figura 6 mostra l'arquitectura bàsica del sistema.

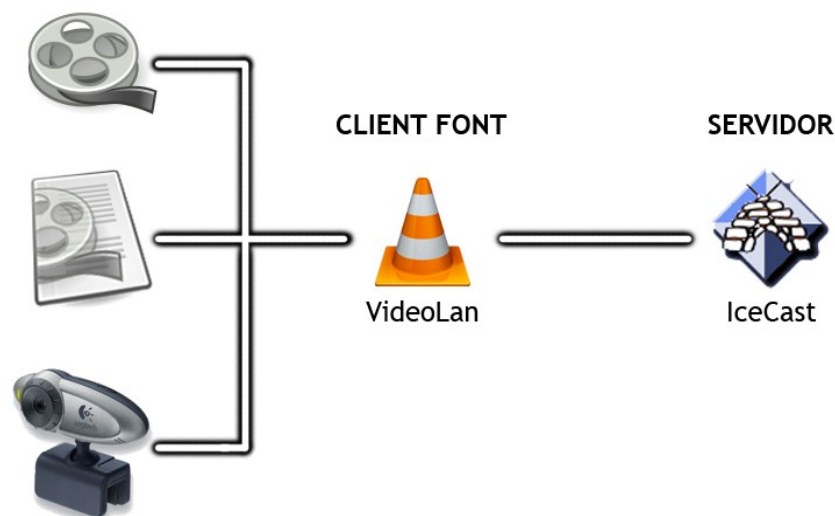


Figura 6: Esquema de corrents de vídeo

5.2.1 Enviament d'un fitxer

Per repetir aquesta combinació però amb vídeo, només cal replicar la mateixa estructura bàsica que amb àudio. En aquest cas el VLC no va oferir bons resultats a l'hora de transcodificar a Theora en temps real, així que va ser necessari adaptar els fitxers amb aplicacions externes abans d'obrir-los amb el VLC. Per transcodificar de MPG o AVI a OGG el programari *ffmpeg2theora* ha donat bons resultats, amb l'excepció dels fitxers MP4. En aquest cas encara es va afegir un pas previ, transcodificant-los primer a MPG amb el *mencoder* i després a Theora amb el *ffmpeg2theora*.

```
./vlc -vvv /home/.../video.ogg --sout '#standard{access=shout,mux=ogg, url=source:***@XXX.XXX.XXX.XXX:8000/vlc_video_ogg.ogg}'
```

5.2.2 Enviament cíclic d'una llista de reproducció

La llista la crearem de la mateixa manera que en el cas d'àudio, però s'afegiran els fitxers de vídeo en OGG. La reproducció cíclica d'una llista de reproducció va ser la demostració preparada amb continguts de Vilaweb TV extrets del seu servei de podcast amb format MP4.

```
./vlc -vvv /home/.../llista_ogg --sout '#standard{access=shout,mux=ogg, url=source:***@XXX.XXX.XXX.XXX:8000/vlc_video_llista_ogg.ogg}'
```

5.2.3 Càmera web

En aquest cas, el format de vídeo que ens dona la càmera prové del video4linux instal·lat al sistema, per tant abans d'enviar-lo cal codificar el senyal a Theora (vcodec=theo) i establir la taxa de bits (vb=1024). Aquesta taxa de bits no és fixa i es pot canviar depenent de les necessitats.

```
./vlc -vvv v4l:/dev/video0 --sout '#transcode{vcodec=theo,vb=1024}:standard{access=shout,mux=ogg,url=source:***@XXX.XXX.XXX.XXX:8000/vlc_webcam.ogg}'
```

La Figura 7 mostra una exemple de reproducció remota de la càmera web que el Departament de Teoria del Senyal i Comunicacions de la UPC té a l'EUETIT de Terrassa.

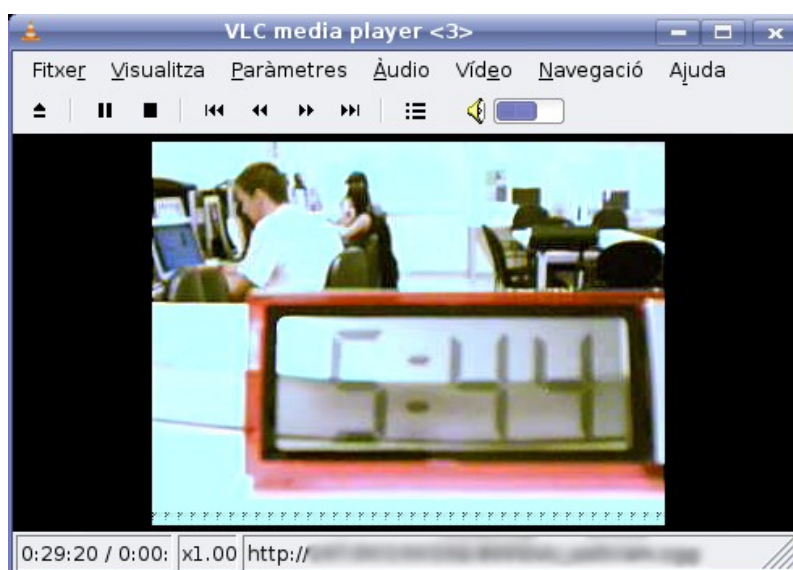


Figura 7: Visualització del corrent de la Webcam

Si es pretén incloure també l'àudio del micròfon, cal codificar ambdues fonts.

```
./vlc -vvv v4l://v4l-vdev="/dev/video0" :v4l-adev="/dev/dsp" --sout '#transcode{vcodec=theo,vb=1024,acodec=vorbis,ab=128}:standard{access=shout,mux=ogg,url=source:***@XXX.XXX.XXX.XXX:8000/vlc_webcam_micro.ogg}'
```

5.3 Reproducció dels corrents

Per a la reproducció dels corrents, utilitzarem el VLC, com a client reproductor. Hi ha dos mètodes bàsics per accedir als corrents disponibles al servidor IceCast: bé accedint a la pàgina web del servidor on s'hi troben enllaços cap a les URLs dels punts muntatge existents, o, si es coneixen a priori, es poden donar directament com a paràmetre d'entrada al VLC.

5.4 Integració dels corrents en una pàgina web

Mitjançant el connector de VideoLan per a Firefox és molt fàcil visualitzar el corrent de vídeo integrat en un document HTML. El següent codi JavaScript n'exemplifica l'ús:

```
<embed type="application/x-vlc-plugin"
  name="corrent"
  pluginspage="http://www.videolan.org"
  id="vlc"
  target="http://XXX.XXX.XXX.XXX:8000/punt_muntatge.ogg.m3u">
<br>
<a href="javascript:;" onclick='document.corrent.play()>Play video</a>
<a href="javascript:;" onclick='document.corrent.stop()>Stop video</a>
</embed>
```

Al departament de Teoria del Senyal i Comunicacions de l'EUETIT, s'ha creat un portal web, dedicat als continguts multimèdia que es generen o proven al departament. Entre els continguts disponibles, es troben els corrents generats amb aquest projecte, en aquest cas la webcam del Laboratori i una llista de vídeos de VilawebTV, que es repeteixen cíclicament, tal com mostra la Figura 8.



Figura 8: Vilaweb TV en una web

6 Conclusions

Una vegada finalitzat el projecte, s'han assolit els requeriments inicials de funcionament total sota programari i formats lliures. Les proves al laboratori han permès comprovar que la solució proposada es manté estable, després de tenir el sistema un temps prolongat funcionant. El projecte ha aconseguit transmetre corrents en temps real d'àudio i vídeo des de tota mena de fonts possibles, arxius, llistes de reproducció, entrada

de la tarja de so i càmera web.

Amb tot, l'arquitectura client – servidor no és eficient a l'hora de transmetre corrents en directe ja que cada nou client significa una disminució de l'ample de banda total del servidor per repetir l'enviament d'un contingut idèntic.

Esforços futurs es centraran en la cerca i desenvolupament de programari lliure que sigui capaç d'utilitzar una arquitectura P2P que eviti la sobrecàrrega del servidor. En la nostra tasca no s'ha aconseguit trobar encara cap solució de qualitat i lliure que solucioni aquest problema, sent el futur projecte IceShare una iniciativa prometedora en aquesta direcció.

Bibliografia

1. Stefano Mosca, "Live streaming and P2P television in the Internet", Projecte de Final de Carrera a la FIB/EUETIT (UPC), 2007.
2. <http://www.xiph.org>
3. <http://www.videolan.org>
- 4: X.Albà, L.Gomes, X.Giró, "Transmissió de vídeo amb programari lliure sobre xarxes IP", IV Jornades de Programari Lliure, Vilanova i la Geltrú, Juliol 2005.