

Web radio con Linux

Il webcasting sta diventando un fenomeno sempre più alla portata di tutti: in questo articolo vedremo come essere "on the air" in modo semplice e veloce utilizzando unicamente Software Libero

Avete mai sognato di creare una stazione radio tutta vostra? Con il Software Libero oggi questo non solo è possibile, ma è anche relativamente facile ed efficiente. In questo articolo vedrete come interagiscono tra loro i vari software necessari allo scopo, e quante possibilità si aprono grazie alla flessibilità di questi pacchetti software. Con streaming si intende un modo diretto di diffondere contenuti audio e video, senza bisogno che lo spettatore/ascoltatore del programma debba scaricare il file, ovvero il contenuto è fruito nel momento stesso in cui viene consegnato dal medium attraverso la rete. Lo streaming audio via rete (LAN o Internet), di cui ci occupiamo in questo articolo, consiste di tre componenti fondamentali (vedi Fig.1):

- la sorgente audio dello stream (o "source client");
- il server di streaming (detto anche "repeater");
- il riproduttore audio ("player" o anche "stream listener").

Analizzeremo l'installazione e la configurazione di un server (Icecast2), di due source client (Ices2 e Darkice), e di Darksnow, un front-end grafico per Darkice. Cominciamo dal componente centrale, il cosiddetto server di streaming, ovvero quel programma che si occupa di dialogare con gli altri due "anelli" della catena: i "source client" e i "listener". I "source client" sono client che si connettono al server per fornire contenuti, mentre i "player" o "stream listener" sono i normali riproduttori audio (XMMS, Alsaplayer, Totem, ecc.) che gli utenti finali useranno per ascoltare lo stream.

L'autore

Emiliano Grilli

Lavora da anni con il Software Libero in campo multimediale, in particolare per quanto riguarda audio e musica. Il suo sito è www.emillo.net.



Installare Icecast2

La nostra scelta è ricaduta sul server Icecast2 (www.icecast.org), poiché si tratta un progetto libero e ben documentato, compatibile con parecchi source client e attivamente sviluppato nonché relativamente semplice da “mettere su strada”. Icecast2 può gestire più di un broadcast contemporaneamente, siano essi diversi stream o versioni diverse (tipicamente a diversi bitrate) dello stesso stream. Ognuno di questi stream deve avere il suo “mountpoint”, ovvero un nome univoco che lo identifica all'esterno. Torneremo su questo più tardi, per ora vediamo come installare il server Icecast2. Dopo aver scaricato il sorgente della versione stabile decomprimetelo con il comando `tar -zxvf icecast-2.2.0.tar.gz` Icecast2 necessita di alcune librerie esterne per poter funzionare, in particolare **libxml2** (per il parsing dei file di configurazione), **libxslt** (per il rendering delle pagine amministrative), **libvorbis** (per l'encoding dello stream) e **libcurl** (opzionale, serve se volete che

la vostra stazione radio sia elencata nelle directory YP: Yellow Pages). Assicuratevi quindi di avere queste librerie installate, da sorgente o con i pacchetti specifici per la vostra distribuzione (in questo caso è necessario installare i pacchetti `-dev` o `-devel` per poter compilare Icecast). Entrate nella directory dei sorgenti e digitate i soliti comandi per la configurazione e la compilazione:

```
cd icecast-2.2.0
./configure
make
```

Se tutto è andato bene potete procedere all'installazione sul sistema, impartendo da utente root il comando `make install`

Il file di configurazione del server

A questo punto dovrete trovarvi con un file di configurazione d'esempio in `/usr/local/etc/icecast.xml`, un file XML molto ben commentato e la cui sintassi è ampiamente descritta nella documentazione di Icecast. Vediamo nel dettaglio i parametri più importanti del file di

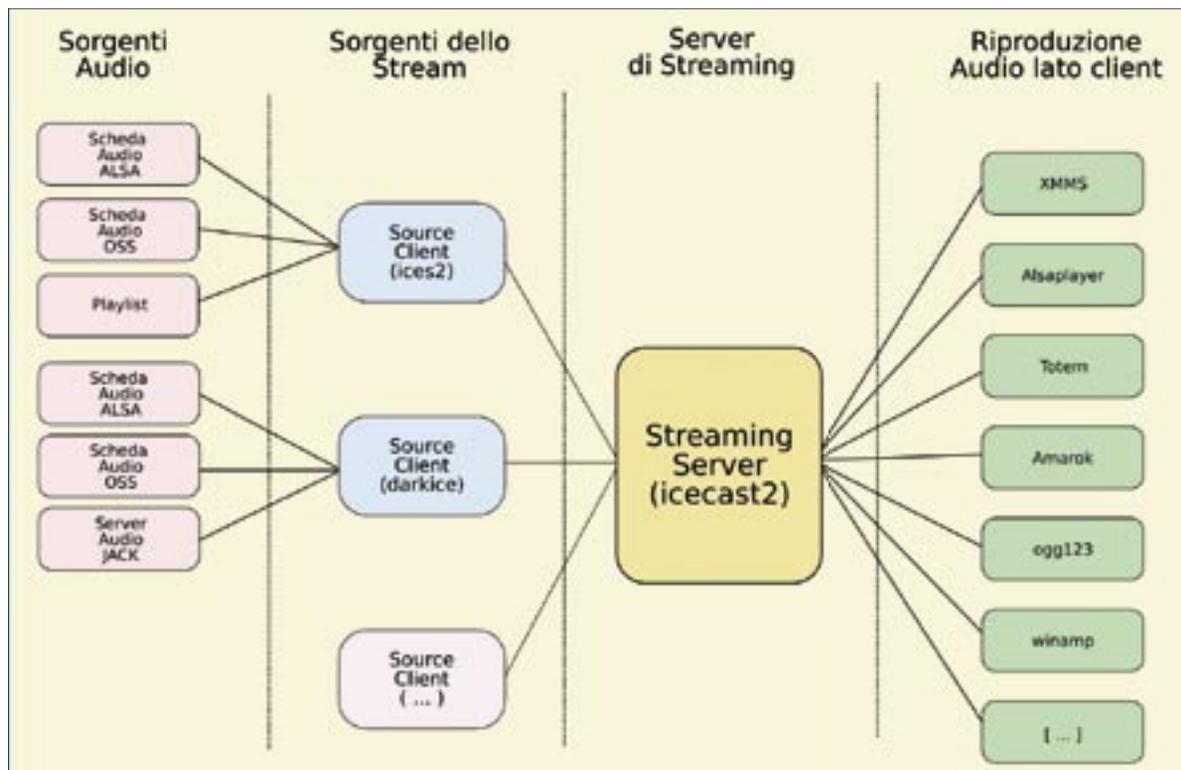
configurazione:

- nella sezione **<limits>** vengono impostati i limiti per il server, ad esempio il numero massimo di source client (parametro `<sources>`, di default impostato a 2), e il numero massimo di ascoltatori (parametro `<clients>`, di default impostato a 100) che il server Icecast2 deve servire. È bene notare che questo limite si riferisce al server nel suo complesso, non relativamente ai singoli mountpoint. Potete lasciare questi valori alle impostazioni predefinite, per cominciare;
- nella sezione **<authentication>** impostate le password per collegarsi al server: il tag `<source-password>` controlla la password utilizzata dai source client per collegarsi al server (con username “source”); il tag `<relay-password>` invece stabilisce la password per i relay, ovvero altri server di streaming per i quali possiamo fare da “ponte” (in questo caso lo username è “relay”). Infine, potete impostare la password amministrativa che consente



di accedere all'interfaccia Web di gestione di Icecast: potete definire il nome utente tramite il tag `<admin-user>`, e la relativa password mediante il tag `<admin-password>`. Un altro tag molto importante da impostare è `<hostname>`, che rappresenta il nome (o l'indirizzo IP) che gli utenti del nostro stream riceveranno all'interno delle playlist generate dal server. Se ad esempio lasciate “localhost”, le vostre playlist sarebbero accessibili soltanto dalla macchina locale. È bene quindi, se volete diffondere la vostra musica anche all'esterno, impostare qui un valore valido che identifichi la vostra macchina sulla rete (sia essa la “big Internet” o una LAN);

- nella sezione **<listen-socket>**, il parametro `<port>` indica la porta TCP/IP alla quale il server Icecast



1 Le varie componenti di uno streaming radio via Internet

Una bussola per orientarsi

Persi fra la pletera dei formati con cui vengono diffuse le radio su Internet? Ecco un riepilogo dei diversi formati, ciascuno seguito dalla lista dei programmi con cui potete ascoltarlo.

TABELLA FORMATI

SHOUTcast	Streamtuner, xmms, Mplayer
Xiph.org (IceCast)	Streamtuner, xmms, Mplayer
Real	Realplayer, Mplayer
Windows	Media Mplayer
Quicktime	Mplayer
Nullsoft	Streaming Video Mplayer



rimane in ascolto: sia i source client che i client si connettono a questa porta, tramite il protocollo applicativo HTTP. Collegandosi tramite un browser alla macchina dove gira il server, specificando la porta scelta nell'URL (ad esempio per la porta 8000 sulla macchina locale: `http://localhost:8000`) si accede a una pagina di statistiche nella quale sono presenti gli stream disponibili e i link per ascoltarli. Aggiungendo a tale URL la directory /admin/ (es. `http://localhost:8000/admin/`) verrà chiesto di autenticarsi come utente amministrativo; fatto ciò si avrà accesso a una serie di pagine che permettono di visualizzare statistiche più dettagliate sui client connessi, aggiornare i titoli degli stream ed effettuare altre operazioni amministrative; segue la sezione <paths>, che serve a specificare i percorsi delle directory usate da Icecast, ad esempio per i file di log e per la directory che funge da "Web root". È importante indirizzare i log su una directory per la quale l'utente che usate per far partire il server abbia i permessi in scrittura, altrimenti Icecast si rifiuterà

di partire ("permission denied"). Per esempio, specificate come directory di logging (parametro <logdir>) la directory /tmp, che dovrebbe essere scrivibile da qualunque utente. In un secondo momento, quando avrete acquisito dimestichezza con il server, potrete creare delle directory ad hoc e far partire Icecast come servizio di sistema (la distribuzione Debian, ad esempio, crea anche un utente dedicato chiamato "icecast"). È bene anche notare che i percorsi che specificherete nella sezione <paths> non devono terminare con uno slash ("/"). Il file di configurazione contiene molte altre impostazioni possibili, ad esempio per la configurazione dei relay, o per impostare la presenza del vostro streaming server all'interno di directory apposite (chiamate "Yellow Pages") che elencano tutte le Web radio che desiderano apparire in questi elenchi. Per ora ci accontenteremo di questi settaggi base e vi lasciamo il piacere della scoperta di questi argomenti avanzati sulla esauriente documentazione di Icecast. Copiate il file di configurazione icecast.xml in una directory accessibile al vostro utente (ad esempio nella vostra home):
`cp /usr/local/etc/icecast.xml ~/`
Ora con il vostro editor di testi preferito potete modificarlo, andando a cambiare secondo le vostre esigenze i parametri che abbiamo descritto. Una volta salvato il file, potete finalmente lanciare Icecast2 indicando sulla linea di comando (con lo switch -c) il file di configurazione da utilizzare:
`/usr/local/bin/icecast2 -c ~/icecast.xml`
A questo punto il server dovrebbe essere in esecuzione, e puntando un browser all'indirizzo `http://localhost:8000` (ipotizzando che abbiate lasciato l'impostazione predefinita per la porta TCP/IP, 8000) dovreste vedere la pagina prodotta da Icecast, chiamata "Icecast status page" (come è visibile in Fig.2), che come noterete ha un aspetto molto simile al sito Web di Icecast2. Qui saranno elencati gli stream

disponibili all'ascoltatore. Vediamo ora come fornire il contenuto degli stream al vostro server Icecast2.

I "source client"

Come abbiamo detto all'inizio di questo articolo, un altro componente essenziale nella catena dello streaming è il "source client", ovvero un programma che vi permette di fornire ad Icecast2 il contenuto audio da servire agli ascoltatori. Abbiamo scelto due source client facili da usare e potenti: Ices2 e Darkice. Il primo è sviluppato dagli stessi autori di Icecast2 e permette di utilizzare tre tipologie di input audio: OSS, ALSA e playlist (una serie di file ogg preregistrati). Il secondo analogamente fornisce ingressi per ALSA e OSS, ma è in grado di dialogare anche con il server audio JACK, permettendo di ottenere una grandissima flessibilità di routing audio tra applicazioni diverse e possiede, a differenza di Ices2, una comoda interfaccia grafica chiamata Darksnow.

■ Installare Ices2

Anche Ices2 ha bisogno di alcune librerie per poter essere compilato: libxml2, libogg, libvorbis e libshout2. Anche in questo caso potete installare queste librerie da sorgenti o usare le versioni incluse nella vostra distribuzione, avendo cura di installare i pacchetti di sviluppo relativi (di solito il loro nome termina con -dev o -devel). In particolare, libshout2 si può scaricare sempre

da www.icecast.org, e fornisce le funzioni di libreria per scrivere applicazioni che dialoghino con Icecast2. Altro requisito, se volete utilizzare l'interfaccia ALSA come metodo di input per Ices2, sono i driver (inclusi ormai nella quasi totalità delle distribuzioni), la libreria ALSA e i file di ALSA per lo sviluppo. Scompattate i sorgenti di Ices2 (reperibili sempre su www.icecast.org oppure sul CD/DVD allegato alla rivista) con il comando:

```
tar -zxvf ices-2.0.1.tar.gz
```

quindi entrate nella directory creata dalla decompressione e impartite i comandi per la configurazione e la compilazione dei sorgenti:

```
./configure
```

```
make
```

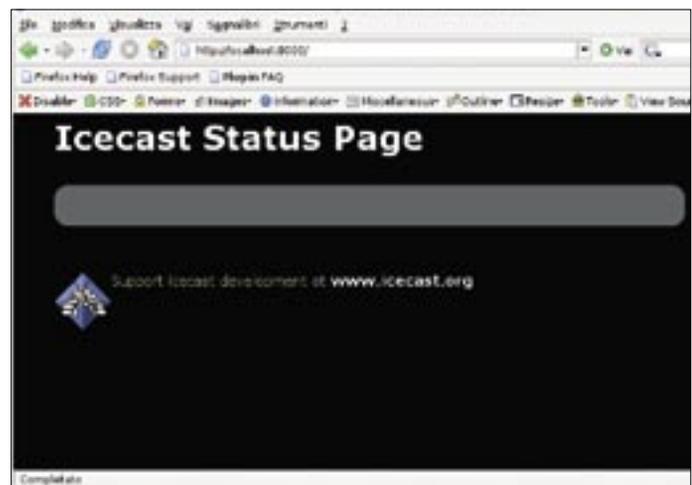
Una volta terminata con successo la compilazione, installate Ices2 con il comando

```
make install
```

da utente root.

■ Configurare Ices2

Analogamente ad Icecast2, anche Ices2 viene gestito tramite un file di configurazione XML: ve ne sono alcuni di esempio nel pacchetto sorgente, ognuno adatto a sfruttare Ices2 con i suoi diversi metodi di input audio. Avete essenzialmente due opzioni: passare ad Ices2 l'input della scheda audio (mediante i moduli di input OSS e ALSA) oppure fornirgli una playlist di file ogg preregistrati. I file di esempio relativi a queste differenti



2 La pagina di stato del server Icecast2, appena lanciato

configurazioni del source client sono: ices-oss.xml, ices-alsa.xml e ices-playlist.xml. Ora copiate nella vostra home il file ices-alsa.xml, che userete per configurare Ices2. Anche in questo caso la documentazione a corredo è molto dettagliata, citiamo qui alcune impostazioni che possono aiutarvi ad iniziare: il tag <background> ad esempio, controlla se il programma debba essere eseguito in background. Per il momento è meglio lasciare questa impostazione a "0". Le impostazioni relative al logging vanno configurate in modo che l'utente che lancia il programma abbia i permessi di scrittura sulla directory specificata dal tag <logpath>. Potete per esempio

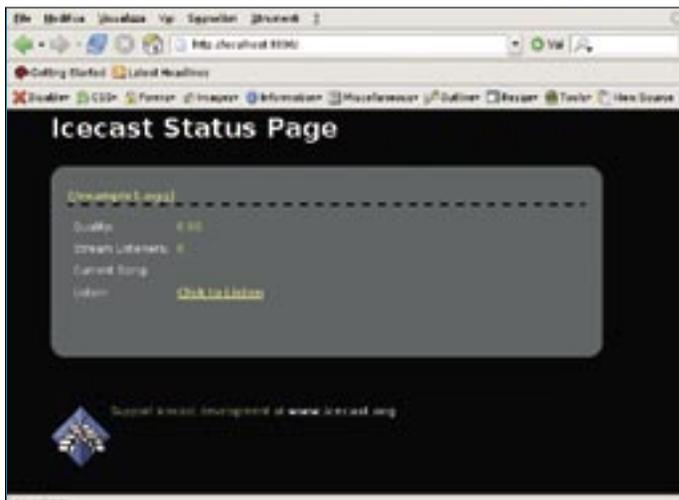
insieme allo stream, come <name>, <description>, ecc. Sempre all'interno di <stream> trovate l'importante sezione <input>, che determina il modo di funzionamento di Ices2 e i parametri per configurare l'input dei dati audio. Qui vedrete che <module> è impostato al valore "alsa", dato che abbiamo preso a modello il file "ices-alsa.xml". Seguono una serie di tag <param> che vi permettono



di specificare i dettagli del vostro stream audio. I settaggi di default per tali parametri di solito sono già sufficienti per iniziare, ma potreste aver bisogno di cambiarli. Ad esempio potreste voler cambiare il "sampling rate" (frequenza di campionamento) da 44100 Hz a 48000 Hz se possedete una Sound Blaster Live! In questo caso sostituite a "<param name='rate'>44100 </param>" la stringa "<param name='rate'>48000 </param>".

Un altro importante parametro è "device", con il quale si sceglie quale periferica audio usare: il default in questo caso è "hw:0,0" e identifica la prima scheda audio ALSA presente nel sistema. Se per esempio avete due schede audio e volete usare la seconda, non dovrete far altro che cambiare questa impostazione in "hw:1,0". Per avere un elenco delle schede audio presenti nel sistema e la loro posizione, usate il comando **cat /proc/asound/cards**. Il tag <channels> imposta il numero di canali audio in ingresso. Dopo <input>, trovate la sezione <instance>, che rappresenta la configurazione della connessione al server Icast2 e alcune impostazioni per la recodifica audio dello stream.

Questa è la parte più critica della configurazione di Ices2. Se riscontrate problemi, probabilmente questo è il primo posto in cui guardare per cercare eventuali errori. La sezione <instance> serve a istruire Ices2 su come trovare il server, come autenticarsi e come effettuare l'encoding dell'audio. Impostate tramite il tag <hostname> il nome o l'indirizzo IP della macchina su cui gira il server Icecast2 (non deve essere necessariamente lo stesso computer...): se state facendo girare il server in locale, potete tranquillamente lasciare il preimpostato "localhost". Configurare successivamente i parametri <port> e <password> in modo che rispecchino quelli che avete impostato nella configurazione del server (i default sono "8000" per la porta e "hackme" per la password). Con il tag <mount> si specifica il mountpoint del vostro stream: è importante, per compatibilità con alcuni player, che questo nome termini con l'estensione .ogg. Questo nome identificherà lo stream sulla rete, ed è quello che gli ascoltatori forniranno al proprio player per ascoltarlo (ad esempio **http://miosito.com:8000/example1.ogg**, dove **miosito.com** è l'indirizzo Internet o LAN del server Icecast2, ed **example1.ogg** il nome del mountpoint). Seguono la sezione <encode> e la sezione <resample>, che determinano il modo di Ices2 per



3 La pagina di stato del server Icecast2 dove appare lo stream di ices2 (example1.ogg)

creare una directory chiamata "iceslogs" nella vostra home, e istruire Ices2 a scrivere i log in tale directory. Inizialmente tuttavia può essere comodo impostare il parametro <consolelog> a "1": in questo modo i log invece di essere scritti sui file specificati vengono scritti direttamente sullo standard output del terminale da cui lanciate Ices2, consentendo una migliore diagnostica dei problemi. Successivamente, quando siete sicuri che tutto funzioni come dovrebbe, potete reimpostare a "0" questo settaggio. Nella sezione <stream> del file di configurazione trovate le impostazioni relative al vostro stream: ad esempio <metadata>, che contiene una serie di informazioni che verranno fornite agli ascoltatori

Glossario

- **bitrate**: la frequenza alla quale i bit transitano sulla rete. Si misura in bit per secondo, e i relativi multipli: kilobit per secondo (kbps), Megabit per secondo (Mbps), ecc. Viene utilizzato per misurare l'occupazione di banda degli stream delle Web radio: valori tipici sono 24 kbps, 32 kbps, 64 kbps e 128 kbps.
- **broadcast**: teletrasmissione, programma radiofonico o televisivo, in generale trasmissione nella tipica configurazione "uno a molti".
- **codec**: contrazione di "Compressor-Decompressor", programma o periferica in grado di comprimere e decomprimere stream di dati in determinati formati di compressione, come ogg/vorbis, mp3, ecc.
- **JACK**: acronimo per "JACK Audio Connection Kit", un server audio a bassa latenza ottimizzato per prestazioni musicali professionali. Permette l'interconnessione audio e lo scambio di stream audio tra applicazioni diverse.
- **ogg/vorbis**: codec aperto e libero che venne proposto in sostituzione del famoso MP3 quando la Fraunhofer Gesellschaft

- (detentrici del brevetto sul codec MP3) annunciò che avrebbe imposto una tassa per l'utilizzo del suo codec. A fronte di una dimensione leggermente maggiore, i file ogg offrono inoltre una qualità audio superiore.
- **player**: programma utilizzato dall'utente finale per ascoltare lo streaming. Tipici player per Linux sono XMMS, Alsaplayer, Totem, Rythmbox, amarok, ecc.
- **sample rate**: frequenza di campionamento. Quando l'audio viene digitalizzato, indica il numero di campioni (sample) utilizzati per ogni secondo di audio digitale. Si misura in hertz (Hz) o kilohertz (kHz). Frequenze comuni sono 48000 Hz, 44100 Hz, 22050 Hz, ecc.
- **source client**: programma che fornisce lo stream audio al server di streaming.
- **stream**: letteralmente "flusso", indica l'oggetto del Webcasting: il flusso audio o video che viene trasmesso.
- **Webcasting**: trasmissione via Web paragonabile a un broadcast televisivo o radiofonico.



codificare l'audio da mandare al server. Vi è anche un tag `<downmix>` che indica ad Ices2 di miscelare i canali in ingresso, solitamente due, in un unico canale monofonico. Se decidete di settare `<downmix>` a "1", e quindi di utilizzare questa feature, bisognerà usare "1" come numero di canali dentro `<encode>`. All'interno di `<encode>` impostate anche la qualità dello stream (tag `<quality>`, in un range da 0 a 10) e la frequenza di campionamento (tag `<samplerate>`): è molto importante che queste impostazioni siano conformi a quanto specificato in `<resample>`, dove si decide come trasformare la frequenza di campionamento

in ingresso (`<in-rate>`) in quella effettivamente usata da Ices2 verso il server (`<out-rate>`). L'operazione di resampling è abbastanza delicata, e di solito per facilitarla si usano valori in cui la frequenza di ingresso sia multipla di quella di uscita (ad esempio `in-rate: 44100 / out-rate 22050`, oppure `in-rate: 48000 / out-rate: 24000`).

Prime prove tecniche di trasmissione

Ora che avete configurato il source client, potete procedere al vostro primo collegamento con il server. Lanciate Ices2 dando come parametro a linea di comando il file di configurazione appena modificato:



4 La schermata di Alsamixer che mostra i controlli di registrazione (View: Capture)

ices2 ices-alsa.xml

Se tutto ha funzionato a dovere, dovrete vedere l'istanza di Ices2 nella pagina di stato del server (nel nostro esempio precedente, `http://localhost:8000`), come mostrato in **Fig. 3**. A questo punto siete "on the air" ed è possibile collegarsi al vostro server di streaming cliccando sul link "Click to listen" che appare nella pagina succitata. Questo link punta a una playlist (in formato .m3u) che contiene l'indirizzo dello stream, e che dovrebbe essere automaticamente gestita dal vostro browser in modo da essere aperta da un'applicazione opportuna (un player audio, di solito XMMS, Alsaplayer, Totem, amarak, ecc.). Ora il mondo è pronto ad ascoltare quanto passa dalla vostra scheda audio, e se siete dei provetti DJ con tutta l'attrezzatura del caso potete collegarla direttamente all'input della scheda sonora. Ma supponiamo che non abbiate sorgenti audio esterne da collegare: è possibile scegliere, tramite ALSA, quale ingresso sia al momento in stato "capture", ovvero cosa stiamo "registrando". Una delle utility più comode per gestire il mixer di ALSA è alsamixer (fornito con le **alsa-utils**). Lanciate alsamixer in un terminale e premete il tasto `<TAB>`: questo vi permetterà di accedere ai controlli relativi alle periferiche di registrazione. Qui avete la possibilità di scegliere quale sia la periferica selezionata per il "capture", e di impostare il volume generale di registrazione. Naturalmente quello che vedrete qui dipenderà molto dal tipo di hardware di cui siete in possesso; alcune periferiche comuni a molte schede sono line, mic, CD, video, ecc. Per impostare la registrazione è sufficiente posizionarsi con i tasti freccia sul canale desiderato e premere la barra spaziatrice. Assicuratevi che il canale "capture" sia selezionato e abbia il volume alzato (vedi **Fig. 4**). Accertatevi inoltre che i canali che volete trasmettere non siano in *mute* (disattivabile con il tasto "M" in alsamixer, dopo esservi posizionati sul canale desiderato). Nella

I server di streaming alternativi

Icast2 è un server di streaming abbastanza semplice ed è uno dei più maturi per quanto riguarda l'audio, ma esistono alternative molto interessanti, come ad esempio **Flumotion** (www.flumotion.net), un server Open Source con una serie di feature molto più ampia, e di conseguenza con un'architettura molto più complessa, e un'interfaccia grafica avanzata che ne rende l'amministrazione più semplice. Flumotion è un progetto molto più giovane ed è attualmente in fase di frenetico sviluppo. Al momento non è realmente utilizzabile, ma è un software sicuramente da tenere d'occhio nel prossimo futuro. Un software molto simile ad Icast2 è invece **Shoutcast**, prodotto freeware di NullSoft, gli autori

di Winamp, reperibile sul sito www.shoutcast.com, orientato soprattutto allo streaming in formato MPEG3. Vi è inoltre un'alternativa commerciale proposta da Real Networks, gli autori del celeberrimo Real Player: il server **Helix** (www.realnetworks.com/industries/resources/technology/helix.html), che permette lo streaming di numerosissimi formati, a partire da RealAudio e RealVideo, passando per MPEG4 e Windows Media. Un altro server alternativo è **Darwin Streaming Server**, o **DSS** (<http://developer.apple.com/darwin/projects/streaming/>), prodotto da Apple e rilasciato sotto licenza APSL (Apple Public Source License), e orientato allo streaming video nei formati quicktime e MPEG4.



5 La finestra "server options" di Darksnow

maggior parte delle schede audio di grado "consumer" è presente un controllo di registrazione chiamato "Mix", che corrisponde al mix globale della scheda, in pratica quello che state sentendo. Selezionando "Mix" come periferica di registrazione potete ad esempio parlare con il microfono sopra una base musicale in esecuzione con XMMS, e il tutto viene trasmesso come stream al server. Attenzione, però! Se usate la stessa macchina sia come source client che come listener, e avete selezionato "Mix" come sorgente di registrazione, state creando un loop audio (l'input rilegge sé stesso) che risulterà in un suono abbastanza sgradevole. In effetti il source client e il listener sono pensati per risiedere in macchine diverse, se volete provare ad ascoltare il vostro stream dalla stessa

macchina dove gira Ices2 è meglio usare un altro modulo di input, ad esempio il modulo playlist, che può essere configurato a partire dal file di esempio "ices-playlist.xml". Fondamentalmente si tratta di istruire Ices2 a leggere un semplice file di testo (una playlist) dove ogni riga rappresenta un file oggi preregistrato. Le righe devono essere almeno due ed essere diverse fra loro. Questo approccio presenta un vantaggio rispetto al passare dalla scheda audio: il vostro stream applicherà automaticamente come titolo della canzone corrente il titolo che trova nel tag ID3 del file ogg. Gli svantaggi sono però che avrete una minore flessibilità sonora (non potrete ad esempio mixare un microfono o un'altra sorgente), e che sarete limitati ai soli file ogg. Per avere un vero "studio radiofonico" nel vostro PC,

introduciamo un altro source client, che ha la particolarità di supportare il potente e flessibile server audio JACK: Darkice.

Installare Darkice/Darksnow

Installare darkice (<http://darkice.sourceforge.net>) non dovrebbe essere molto complicato, ora che avete tutte le librerie e i file per lo sviluppo (in particolare, Darkice necessita di libogg e libvorbis per fare l'encoding dello stream, ed eventualmente la libreria lame per creare stream in formato MP3). Avrete anche bisogno del server audio JACK e delle librerie relative, per usare questo potente modulo di input per Darkice. Dopo averlo scaricato o copiato dal CD/DVD allegato alla rivista, procedete come al solito alla decompressione, alla configurazione e alla compilazione del sorgente:

```
tar zxvf darkice-0.15.tar.gz
cd darkice-0.15
```

```
./configure
```

```
make
```

e quindi all'installazione sul sistema; da utente root digitate:

```
make install
```

Come dicevamo, per controllare Darkice esiste una comoda interfaccia grafica GTK2 chiamata **Darksnow** (<http://darksnow.radiolive.org/index.en.html>).

Darksnow è un programma piuttosto semplice e non dovrete avere problemi a compilarlo: ricordate solo che sono necessarie per la compilazione le librerie e i tool di sviluppo GTK2.

Procedete come al solito:

```
tar zxvf darksnow-0.5.2.tar.gz
```

```
cd darksnow-0.5.2
```

```
./configure
```

```
make
```

e quindi installatelo nel sistema come utente root:

```
make install
```

Nel caso non voleste per qualche motivo installare le librerie di sviluppo per GTK2, sul sito di Darksnow l'autore mette a disposizione anche pacchetti binari per alcune distribuzioni.

Configurare Darkice/Darksnow

Ora potete lanciare Darksnow



Dyne:bolic, il live streaming tutto italiano

Un progetto che non potevamo non menzionare in un articolo sullo streaming audio è il Live CD Dyne:bolic (<http://dynebolic.org>), che offre una piattaforma "out of the box" per lo streaming e la produzione di contenuti multimediali. Inoltre è un progetto portato avanti soprattutto da sviluppatori italiani. Dyne:bolic include, oltre al server Icecast2, anche un source client grafico prodotto dagli stessi autori della distribuzione, chiamato **MuSE** (acronimo per "Multiple Streaming Engine"), tramite il quale è possibile miscelare file musicali con le sorgenti della scheda audio (ad esempio un microfono). Il Window Manager scelto da Dyne:bolic è Window Maker. Per cominciare lo streaming è sufficiente cliccare sul desktop col pulsante destro del mouse e scegliere **AUDIO** ⇒ **STREAM** ⇒ **Icecast2:: start streaming server** dal menu di Window Maker. A questo punto si aprirà un terminale di controllo per

il server (se volete terminare il server premete CTRL+C nel terminale). Una volta messo in esecuzione il server, è possibile lanciare MuSE dal menu (**AUDIO** ⇒ **STREAM** ⇒ **MuSE:: radio streaming**). Cliccando sulla prima icona in alto a sinistra (che assomiglia a una spina inserita in una presa) si apre una finestra dove impostare le opzioni per la connessione al server e i parametri audio dello streaming (che viene offerto sia in formato MP3 che ogg). Cliccando con il pulsante destro nell'area bianca centrale invece è possibile aggiungere file audio per lo streaming. Dyne:bolic offre anche l'interessante possibilità di salvare i dati delle proprie sessioni (in particolare i file di configurazione delle applicazioni) con un procedimento chiamato "nesting", su una partizione esistente o una chiave USB. Sono necessari almeno 32 MB per questa operazione, descritta nel wiki di dyne:bolic all'URL <http://lab.dyne.org/Nesting>.

di banda. Ovviamente questi due indicatori sono inversamente proporzionali, si tratta soprattutto di trovare il giusto compromesso per il programma che state trasmettendo. Ad esempio se intendete trasmettere del semplice parlato ha poco senso usare la stereofonia, molto meglio usare un canale solo (mono), e una frequenza di campionamento bassa (22050 Hz o 11025 Hz). Altro aspetto da considerare è la banda a disposizione e il numero di "stream listener" che volete supportare contemporaneamente. Ad esempio se state lavorando in una LAN per creare una Webradio aziendale, non avete problemi di banda e potete usare la qualità massima per lo stream, ma se invece siete su una connessione ADSL e volete fare streaming di una conferenza su Internet, converrà che limitiate il più possibile la banda utilizzata da ogni listener connesso, in modo da poterne servire un numero maggiore. L'ultima casella della linguetta "audio options" è intitolata "device input" ed è dove sceglierete il modulo di input di Darkice. Qui trovate le periferiche OSS (/dev/dsp*), ALSA (hw:x,y)

(Fig.5): la prima linguetta, chiamata "server options", vi permette di configurare i parametri di connessione al server Icecast2 (che naturalmente deve essere in esecuzione). Impostate il parametro "Server" su "localhost", nel caso che Icecast2 stia girando sulla stessa macchina da cui vi state connettendo, oppure immettete qui l'URL o l'indirizzo IP del server Icecast2 remoto. Nella casella "Port" immettete il valore (il default è 8000) della porta su cui il server è in ascolto. Immettete nella casella "MountPoint" un nome che identifichi il vostro stream, ad esempio "mywebradio.ogg". Ricordatevi che questo nome deve terminare con l'estensione ".ogg" per ragioni di compatibilità verso alcuni client. Immettete la password che avete impostato per la connessione al server Icecast2 nella casella "password". La seconda linguetta (Fig.6) dell'interfaccia di Darksnow è chiamata "audio options", nella quale configurerete il tipo di stream, le impostazioni sul formato e la qualità audio, e il tipo di input che volete fornire a Darkice. Darkice (e di conseguenza Darksnow) supporta anche il formato MP3 per lo streaming: per ottenere questa feature occorre compilare Darkice con il supporto MP3 fornito dalla libreria lame, che non è libera in quanto il codec MP3 è protetto da un brevetto che ne limita l'utilizzo. Scegliete pertanto "ogg/vorbis" nella casella "Format" per avere

il vostro stream in un formato libero e aperto, e oramai compatibile con tutti i player. Gli altri parametri nella linguetta "audio options" servono a stabilire le caratteristiche del formato audio del vostro stream: il bitrate (in kbps - vedi glossario), se il bitrate debba essere costante o variabile, la frequenza di campionamento, il numero di canali, ecc. Giocate con questi parametri finché non trovate una combinazione soddisfacente di qualità audio/occupazione



6 La finestra "Audio Options" di Darksnow



7 Darkice usato come client JACK appare nella finestra "connect" di qjackctl



e le opzioni "jack" e "jack_auto" per connettere Darkice al server audio JACK. Scegliete quest'ultima opzione. L'ultima linguetta di Darksnow (streaming description) riguarda le informazioni testuali usate per descrivere il vostro stream. Queste informazioni verranno passate al player dell'ascoltatore come meta informazioni (titolo, descrizione, URL, genere, eccetera). Scegliete qui "yes" nella casella "public". Questo permetterà a chiunque di ascoltare la vostra radio. La parte bassa dell'interfaccia di Darksnow contiene tre pulsanti: premete subito "show details" per aprire una finestra di log dalla quale potete vedere i messaggi di Darksnow: ciò è molto utile per diagnosticare eventuali problemi. Gli altri due pulsanti sono "Start Streaming" e "Stop Streaming", dal significato abbastanza evidente. Abbiamo scelto come modulo di input "jack_auto", ovvero una interfaccia che permette a Darkice di identificarsi come client verso il server audio JACK. Nella **Fig.7** potete vedere il programma **qjackctl**, un comodo

front-end per il server JACK, con aperta la finestra "connections": qui potete vedere come Darkice appaia come una porta alla quale sono collegate sia le porte "alsa_pcm" (che rappresentano gli input della scheda audio), che le porte JACK del player XMMS.

Interazione con JACK

Dopo aver fatto partire il server JACK (magari utilizzando qjackctl) premete il pulsante "Start Streaming" in Darksnow e sarete in grado in pochi secondi di nutrire il vostro stream con una miriade di applicazioni audio che potete collegare e scollegare a piacimento dalla comoda interfaccia di qjackctl. Per i dettagli su JACK e sulle applicazioni supportate si veda il sito Web del progetto: <http://jackit.sourceforge.net>. JACK, che è stato discusso in precedenza sulle pagine di questa rivista, è ormai sinonimo di audio professionale su Linux, e vi sono già moltissime applicazioni che lo supportano: esso offre una flessibilità enorme di routing audio e avere un source

client come Darkice che lo supporta apre moltissime strade creative. Ad esempio, l'autore di questo articolo ha partecipato alla "Linux audio conference 2005" a Karlsruhe (Germania), dove a scopo dimostrativo è stato utilizzato Darkice per effettuare un "mix a distanza" direttamente dall'uscita JACK di Ardour (un multitraccia professionale) da uno studio di registrazione situato a Minneapolis (USA). Il tecnico del suono negli USA apportava al mix le modifiche richieste dal relatore in Germania e queste potevano essere ascoltate dopo pochi secondi in streaming dal pubblico in sala. Tutta la conferenza è stata inoltre trasmessa in streaming audio e video, generando una discreta audience "virtuale" proveniente da tutto il mondo.

Conclusioni

Speriamo con questo articolo di aver "stuzzicato l'appetito" sull'argomento dello streaming audio via Internet, e di aver dato un'idea delle grandi potenzialità che offre questo mezzo. All'aumentare della banda

il Webcasting diventerà un fenomeno sempre più comune, oltre che per diletto anche per scopi professionali. Si pensi per esempio alla possibilità di offrire via Rete lo streaming di lezioni, conferenze, audiolibri, ecc. Per non parlare degli sviluppi di questa tecnologia legati allo streaming di dati video, finora appannaggio quasi esclusivo di tecnologie proprietarie come RealMedia. Oppure pensate anche all'impatto che questo mezzo può avere sui produttori indipendenti di musica (le cosiddette "indie"), capaci da un giorno all'altro di fornire a bassissimo costo un servizio di promozione di questo genere per i loro prodotti, in tutto il mondo 24 ore su 24, 7 giorni su 7. Questi formidabili strumenti sono, e diventeranno sempre più corredo standard di qualsiasi comunità virtuale che si rispetti, e la loro modularità e flessibilità apre strade pressoché infinite alle menti creative. Giunti a questo punto non possiamo che augurarci che apprezziate il "bello della diretta"!