

Uso di dvswitch per gestire un diretta streaming

dvswitch è un simpatico programmino scritto da alcuni sviluppatori Debian per gestire la diretta streaming della DebConf, la conferenza annuale di Debian. Ha funzionalità molto scarse ed essenziali, ma sufficienti per chi non ha molte pretese e vuole organizzare lo streaming di un evento.

Le istruzioni generali per usare il programma sono sul sito <http://dvswitch.alioth.debian.org/wiki/>, e non mi metterò a ripeterle qui perché sono già abbastanza chiare sul sito (anche se in ordine un po' sparso). In queste pagine aggiungo solo un po' di trucchi che ho scoperto nel tempo per fare cose particolari o con strumentazione particolare.

Breve nota di nomenclatura

Gli esempi mostrati saranno fatti con il programma avconv, che è il nuovo nome del programma ffmpeg. Teoricamente dovrebbero funzionare altrettanto sostituendo ffmpeg al posto di avconv, ma non ci giurerei (anche perché nel tempo l'interfaccia di ffmpeg è cambiata parecchio).

Informazioni generali

dvswitch è un programma GUI che va lanciato con le opzioni -h e -p per specificare rispettivamente un indirizzo (eventualmente 0.0.0.0 che corrisponde a tutte le interfacce) ed una porta sulla quale ascoltare. Una volta avviato, a lui si possono connettere dei client che sono source (se forniscono audio/video) o sink (se lo prendono). Man mano che vengono aggiunte sorgenti, queste compaiono sullo schermo in basso e possono essere mandate in diretta.

Poi ci sono vari programmi dvsource-* e dvsink-* che fanno da sorgenti e pozzi da e verso vari tipi di input (file, telecamere Firewire, sorgenti Video4Linux2, ...). Accettano in input e danno in output solo file e stream in formato DV, che è quello usato interamente da dvswitch.

Come ottenere file DV

Se si usa avconv esiste già il target pal-dv che codifica automaticamente un file in formato DV (usando lo standard PAL,

che è quello che si usa in Italia). Quindi si può fare una roba del genere:

```
avconv -i input.avi -t pal-dv output.dv
```

Il file ottenuto può essere poi dato in pasto a dvsource-file (considerare l'opzione -l per farlo andare in loop).

Come ottenere stream DV da una reflex digitale Canon

Io ho una reflex digitale Canon (modello EOS 500D). Tramite un astuto programmino che si chiama eos-movrec (scritto da un russo e probabilmente non mantenuto, ma quasi funzionante; vedi sito <http://sourceforge.net/projects/eos-movrec/>) è possibile ottenere dalla macchina fotografica uno stream in formato MJPEG (Motion JPEG, ossia tanti frame ciascuno compresso con JPEG appiccicati uno dopo l'altro) incapsulati in un file AVI. Però serve una patch (== hack schifoso) scritta da me (Giovanni Mascellani) per fare in modo che questo stream sia consumabile da avconv: eos-movrec_stream.patch.

Una volta applicata la patch, bisogna creare una FIFO temporanea:

```
mkfifo /tmp/stream.avi
```

Poi bisogna dire a eos-movrec di salvare su tale file (uno degli effetti della patch è fare in modo che non reagisca male quando gli chiedi di salvare su un file che esiste già) e girare l'output della FIFO a avconv:

```
avconv -i /tmp/stream.avi -target pal-dv - | dvsource-file -h host -p port /dev/stdin
```

Attenzione: in questa configurazione un grosso problema è alimentare la macchina fotografica per sufficiente tempo. Tende a scaricarsi piuttosto rapidamente e l'attacco all'alimentazione di rete costa un occhio della testa.

Come ottenere uno stream DV da una telecamera IP

Mi è capitato di dover interfacciare alcune telecamere IP con dvswitch. Non è stato semplicissimo, ma alla fine le forze del bene hanno trionfato.

Tanto per cominciare, bisogna configurarle opportunamente, con particolare attenzione alla configurazione di rete ed a mettere una

password all'interfaccia di amministrazione. Però l'accesso al video deve essere libero da password, perché non so come passare la password a avconv.

L'unico dettaglio a cui fare attenzione è il fatto che avconv non è in grado di capire da solo il tipo di stream (che, per le telecamere con cui ho giocato io, è sempre MJPEG, ma non è incapsulato in un AVI, è sparato direttamente sullo stream). Inoltre è anche una buona idea correggere il framerate e farlo diventare 25 fps (quasi tutte le IP camera che ho provato non fanno più di 15; in realtà potete anche lavorare a 15, ma è meglio che tutte le sorgenti abbiano lo stesso framerate, altrimenti dvswitch può sbarellare e produrre degli artefatti). Bisogna quindi usare il comando leggermente più raffinato:

```
avconv -f mjpeg -i indirizzo_dello_stream -target pal-dv  
-r 25 - | dvsource-file -h host -p port /dev/stdin
```

Ottenere indirizzo_dello_stream può essere un altro problema: alcune volte l'interfaccia Web lo dice; se questo non accade, la pagina http://forum.synology.com/wiki/index.php/User_Reported_Compatible_IP_Cameras potrebbe essere un utile fonte di informazioni.

In alcuni casi la telecamera è particolarmente antipatica e manda fuori stream MJPEG molto buffi, ossia in cui i frame sono divisi da separatori MIME. In questo caso la cosa più facile da fare è mettere un piccolo programmino Python scritto da me (mjpeg_filter.py) per l'occasione che in sostanza separa i frame (che sono normalissime immagini JPEG) e poi li ricuce insieme semplicemente giustapponendoli nello stream MJPEG. Viene fuori una pipeline di questo tipo:

```
curl indirizzo_dello_stream | python mjpeg_filter.py |  
avconv -f mjpeg -i - -target pal-dv -r 25 - | dvsource-  
file -h host -p port /dev/stdin
```

Va da sé che il programmino è hacckosissimo e funziona con la IP camera che avevo io (una Grandtec IP Camera III). Con altre telecamere potrebbe aver bisogno di ulteriori adattamenti (per esempio per riconoscere altri header, o header ordinati diversamente; date un'occhiata al vostro stream con hexdump e quasi tutto dovrebbe essere subito chiaro).

Come ottenere uno stream DV da un dispositivo Video4Linux2 a caso

A volte i dispositivi v4l2 emettono uno stream DV. In tal caso non c'è niente da fare ed è sufficiente usare il programma dvsource-v4l2-dv. È il caso migliore e, ovviamente, a noi non è capitato.

Altre volte avconv è in grado di recuperare lo stream dalla webcam direttamente e di codificarlo e poi passarlo a dvsource-file. In tal caso un comando di questo tipo dovrebbe essere sufficiente:

```
avconv -f video4linux2 -i /dev/video0 -target pal-dv - |  
dvsource-file -h host -p port /dev/stdin
```

In casi disperati, occorre prima passare per vlc, che, a quanto pare, è più astuto nel riconoscere l'input e metterlo in una forma che avconv sia disposto a mangiare. Viene fuori una roba di questo tipo (ammesso che la webcam spari fuori MJPEG; bisogna cambiare il formato se si tratta di altro):

```
vlc v4l2:///dev/video0 --sout  
'#std{mux=raw,access=file,dst=-}' | avconv -f mjpeg -i -  
-target pal-dv - | dvsource-file -h host -p port  
/dev/stdin
```

Questa riga, per ora, ha un problema: la prima esecuzione di vlc introduce un delay di qualche secondo (probabilmente c'è qualche buffer da qualche parte). Devo ancora capire il problema esatto.

Come ottenere uno stream DV dal monitor

Esiste x11grab!

```
avconv -s hd1080 -f x11grab -r 25 -i :0.0 -vf scale  
-target pal-dv - | dvsource-file -h host -p port  
/dev/stdin
```

Purtroppo bisogna specificare per forza le dimensioni dello schermo con l'opzione -s, altrimenti fa il fallback a 640x480...

Sparare tutto ad un server Icecast

Una volta che tutte le sorgenti sono collegate, dvswitch sta funzionando alla grande, ma è del tutto inutile se dello stream non si fa qualcosa di utile. La cosa migliore è ricodificarlo in un formato

opportuno e poi spararlo in uno stream pubblico in modo che tutti possano vedere i vostri magnifici contenuti.

Per fare questo serve un server Icecast (ce ne sono anche altri, ovviamente, ma non li ho provati), che riceve lo stream e lo rende disponibile a tutti quelli che lo vogliono. Una volta installato il pacchetto icecast2, lo si configura (la configurazione di default va già quasi bene, e le modifiche sono piuttosto immediate) e poi si usa il programma oggfwf per mandargli lo stream sorgente.

Nella configurazione sono contenute due password: quella di amministratore (che permette l'accesso all'interfaccia Web di amministrazione) e quella di "source", che serve a oggfwf per autenticarsi come sorgente.

In definitiva, una volta configurato Icecast, il comando che serve è qualcosa del tipo:

```
dvsink-command -h host -p port -- \
  ffmpeg2theora -o - -f dv -x 320 -y 240 -F 25 -V 200 -A
96 --aspect 4:3 --deinterlace - \
  | oggfwf icecast-hostname icecast-port password path
```

Dove password indica la password di sorgente e path il percorso relativo sotto il quale comparirà lo stream (se metti /test.ogg bisognerà chiamare l'URL <http://server:porta/test.ogg> per vedere lo stream). Per sapere il significato delle varie opzioni di ffmpeg2theora, date un'occhiata alla pagina di manuale.

Ovviamente si possono avviare più sorgenti contemporaneamente (usando path diversi, e comunque solo dopo aver configurato opportunamente il numero massimo di sorgenti in Icecast), per esempio una in alta qualità ed una in bassa qualità per chi ha la connessione scarsa. Occhio, però, che ffmpeg2theora potrebbe essere piuttosto pesante.

Qualche suggerimento sparso

Per ridurre il peso dello stream, un accorgimento molto importante e che in genere permette di risparmiare molta banda con spesa relativamente bassa in termini di qualità è ridurre il framerate. Per esempio, usando 15 al posto di 25 la differenza si vede poco (soprattutto se avete input video di qualità non altissima) e la banda quasi dimezza.

Se invece potete permettervi più banda, potete aumentare la dimensione dello stream a 640x480, e alzare il framerate.

Ovviamente dovete anche alzare il bitrate, altrimenti non serve a niente, anzi probabilmente peggiora!

Per visualizzare il video in un browser sfruttando il potentissimo HTML5, date un'occhiata a <http://videojs.com/>.

Ricordatevi anche di salvare su file ciò che inviate in streaming! Bisogna usare dvsink-files e pigiare il pulsante "Record" nell'interfaccia di dvswitch. Fare una sola di queste due cose non è sufficiente: assicuratevi di leggere la documentazione e fare qualche prova prima, perché sarebbe molto spiacevole rendersi conto troppo tardi di non aver registrato nulla.

Per altre informazioni o per condividere esperienze, potete contattarmi all'indirizzo mascellani@poisson.phc.unipi.it (meglio ancora, potete scrivere sulla mailing list del GULP gulp@gulp.linux.it, dopo esservi iscritti sulla pagina <http://lists.gulp.linux.it/mailman/listinfo/gulp>).

Giovanni Mascellani