

## Streaming servers and codecs

# GO WITH THE FLOW

If you want to use Linux to stream video over the Internet, you might have to fight your way through a jungle of codec alphabets. The available software is anything but intuitive. This article will help you pick the options that best fit your needs. **BY OLIVER FROMMEL**

**T**here are two basic approaches to distributing video over the Internet and also two different streaming variants. On top of these choices, there are also video and audio formats to consider. This article will help you decide what best suits your needs.

One video distribution approach is to download the file completely and view the local copy. The other approach is to opt for streaming, which allows users to watch something as soon as the first few packets have reached their machines. The stream can start on demand by the client, or it can start at a fixed time without waiting for the viewers (live stream).

In both cases, the stream provider has

to make a number of technical decisions, like which video and audio formats to use. Depending on the context, this can refer to different things (although people tend to confuse them), such as the video and audio codecs, or possibly container formats. Apple's QuickTime is an example of a container format that can contain different codecs, like Sorensen. The terminology is also inconsistent between technologies. For example, MPEG refers both to the codec and the container format. I think codec is the best choice.

There are a number of important aspects to consider, like the target audience for you streams. If your audience mainly comprises Linux users who will have no trouble installing MPlayer and so forth, you have a large choice of possible codecs.

If you intend to serve Windows and Mac users, both systems have a couple of pre-installed codecs (WMV on Windows and QuickTime on the Mac), but

users will need to manually install anything else.

### Free, Open, Patented

Depending on the kind of use you have in mind, licensing and legal aspects can also be important. If you are just streaming a video to your buddies in your living room from your basement, you can probably ignore licensing issues, but commercial use is an entirely different thing. Even if encoders for various codecs are available free of charge on Linux, actually using them may be risky from a legal standing. For example, MPEG technologies are all patented in some way or another.

Depending on the format, vendors need to license their encoders or decoders with the MPEG consortium [1], and because the licensing fee normally depends on the number of boxes a vendor sells, this approach is not useful for free software developers. The same thing ap-

plies to almost any usable codec; purchasing a commercial encoder software from MainConcept [2], for example, at least keeps you on the safe side.

Real Networks [3] is the veteran in the streaming industry. Real sells encoders, players, and streaming software, but it has also developed its own codecs. As you might expect from a codec that has been under development for ten years, the Real codecs provide excellent video and audio quality in comparison to the bandwidth needed. The server and encoder tools are mature and fairly stable. On the downside, the technology is proprietary throughout, and you need the Real Player client-side. Although Real has placed some of its software under a free license – following the open source trend – this does not include the proprietary codecs.

Real implements the SMIL standard, which is useful for rich media presentations because it can handle streams with different media types by keeping text as text and not converting it to video, for example. The RealProducer encoder will read various video formats and produce Real files from them for different bandwidths. For live streaming, RealProducer reads analog video from V4L devices, but it does not support digital DV over FireWire.

If you would like to try out Real products, you can download the free variants, all of which have some restrictions. For example, the free Helix server only supports 20 simultaneous streams.

Apple has taken a similar approach as Real, publishing a free variant of the Apple QuickTime streaming server under the name of Darwin streaming server [4]. Darwin does not have the QuickTime components, but this does not mean the product is entirely useless; after all, it can still stream MPEG-4, a codec that achieves good results.

In contrast to Real, Apple does not have a portfolio of commercial products for Linux or even an encoder. The standard approach to streaming with the Darwin server is to use the MPEG4IP suite – written by developers at Cisco – for encoding. Using MPEG4IP or the other popular encoders for Linux (MEncoder, Transcode, FFmpeg or VLC) helps you easily create MPEG-4 files from a variety of video formats. MPEG4IP also has

a live encoder that reads analog video from V4L devices and creates MPEG-4 output. This still does not give you an easy solution for handling DV video.

## Flumotion

There is also the free Flumotion server [5] by Fluendo, a company that has been trying to earn money with Linux in the multimedia industry for quite some time. Some of the Fluendo team are the core developers on the GStreamer project, which provides the multimedia infrastructure for GNOME. Fluendo also works on licensed technologies, like DVB and MPEG, and there are plans to release commercial plugins for the Flumotion server in the future.

The free variant still has much to offer for non-trivial streaming projects. Flumotion will handle both V4L and DV input, thus covering most devices. Unfortunately, the server is difficult to install, with dependencies that include the GStreamer, GStreamer Python, python-gtk, and twisted packages in a combination of versions that many distributions will not have by default.

Flumotion itself comprises the Worker and Manager components as well as an administrative front end (Figure 1) along with a setup wizard. The Worker handles encoding and sends video and audio data to the Manager component, which in turn provides connection handling and streaming services. Unfortunately, the Manager needs a different port for each stream, which means that the port number will always change depending on the number of streams the server is currently broadcasting. Flumotion developer, Andy Wingo, has some tips in his blog for live streaming, covering everything from the camera to the software [6].

## Theora

The free Flumotion product only supports a single codec family; on the other hand, this is your only option anyway if

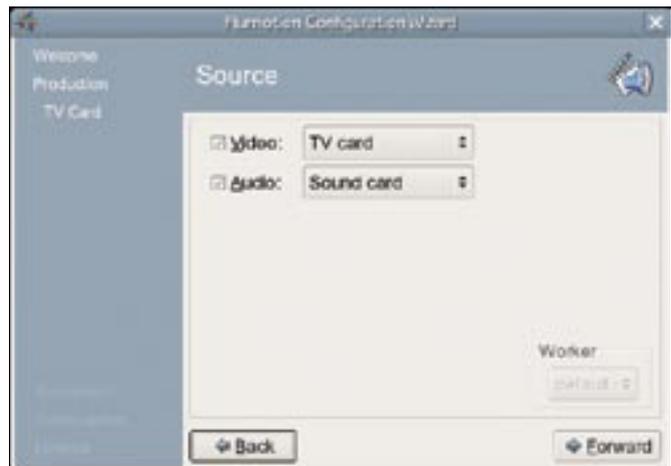


Figure 1: The free Flumotion server uses the Theora format for streaming and supports both analog and digital video.

you are looking for a free solution. Both codecs are from the Xiph Foundation [7], which aims to implement free alternatives for today's patented codecs. Xiph's Vorbis codec for audio achieves a similar quality and encoding efficiency to MP3. You may be more familiar with Vorbis under its container name, Ogg.

The Xiph developers rely on a gift from On2 for the video codec. On2 granted Xiph an irrevocable right of use, which covers any and all patent rights. The product of this union is Theora, which doesn't quite achieve the same quality as MPEG4 or H.264, but it can still produce acceptable results at low bitrates, comparable with Divx 3. Because you can just ignore the legal issues here, almost every distribution will include both codecs.

The Windows and Mac platforms can cause headaches, as neither includes the codecs. There is a Directshow implementation of the Xiph codec for Windows, and the codec will integrate with the Windows codec infrastructure [8]. After the install, the codec is available to all Window video applications. There is a QuickTime component [9] for the Mac.

Fluendo's Cortado applet, a Java-based decoder for Vorbis and Theora, is a good way to avoid most client-side codec issues. The applet works amazingly well, bringing smooth streaming to less powerful systems (with the exception of the Mac, where it looked more like slow motion).

## Icecast

As my own streaming solution, I finally decided on the Icecast server [10]. Ver-

sion 2 can handle Theora streams. One reason that I made this decision was Icecast's ability to accept encoder streams at mount points that are accessible via the same host and port, which is preferable to the tangle of ports that the Flumotion solution requires.

The kind of wizardry that Wingo [6] talks about was unnecessary; a single modification to the Icecast configuration was all it took to prepare the server for Theora streaming. Setting up the Flumotion package posed far greater problems, despite (and in part, because of) the administrative GUI.

`ffmpeg2theora` [11] is a good choice of encoder for Icecast. It can create Theora files from any format that `ffmpeg` can read. If you write the files to `stdout`, the `ogg_fvd` tool can pick them up and send them to the server.

To read live video from a DV camera, I used the `dvgrab` command line tool. This tool provided me with a command line-based, but also understandable encoding pipeline:

```
dvgrab --format raw - | 2
ffmpeg2theora -V 150 -f 2
dv -x 320 -y 240 -o 2
/dev/stdout - | oggfvw 2
IP address port 2
password
Mountpoint.ogg
```

While sad-looking icons in the Flumotion interface show that a component is not working (and who knows why not), users at least get immediate and intelligible feedback using this approach. Most issues are caused by incorrect permission settings for the video device or by server authentication.

Other Linux tools support similar approaches. For example, most transcoders, like `Mencoder`, `Ffmpeg` or `Transcode`, are capable of sending data to `stdout`. `Ffmpeg` can even handle streaming itself, although 20 pages of command line help can be daunting, and much of the online

documentation on the web is now obsolete.

The same thing can be said of VLC, a tool described by many as superior to anything else on Linux. The former `vlc-server` – which many websites still refer to – has now been ousted by the VLC client's streaming option.

The VLC client's GUI is not much help, and even simple command lines can take up multiple lines, not to mention the syntax, with its mix of brackets, braces, and parentheses. Consider whether you prefer an out-of-the-box approach that works or would rather have hours of fun tinkering with a solution that, in the end, might not turn out to be superior after all.

## Protocols

Besides using different codecs, the streaming solutions covered in this article use different network protocols. Thanks to its many years of experience, Real once more has the biggest selection. The server supports multiple protocols, as do the Real clients, so they should have no trouble talking. By default, Real components use the proprietary RDT protocol, although the clients also speak the RTSP (Real Time Streaming Protocol) standard protocol or RTP (Real Time Protocol), which is particularly useful for multimedia data.

Of all the servers covered here, the Darwin Streaming Server is the only one to support RTP/RTSP, whereas all the others rely on HTTP. Unfortunately, HTTP is not ideal for streaming; if a gap occurs due to packet loss, the effect is amplified by the fact that lost packets are retransmitted.

It would be preferable to simply ignore the losses and carry on at the best-possible speeds. To transmit Ogg/Theora in RTP packets, there is no need for the server to implement RTP; although you do need to consider how to encapsulate Ogg in RTP packets. Developers have been working on a payload specification

for a while, but don't hold your breath for results.

## Fun for the Family

Are Linux users ultimately spoiled for choice? Not if you insist on a free streaming solution. Your only option for video streaming is the free Theora codec in combination with Flumotion or Icecast, plus `ffmpeg2theora`. If patents are of no concern, but interoperability is, you might like to investigate the Darwin Streaming Server, which can handle live streams via MPEG4Live. If money is not an issue, you might find the perfect streaming server in the Real product portfolio, and at least there are free (as in beer, if not free as in freedom) clients for any platform you care to name.

Check out Table 1 called "Features" for an overview. ■

## INFO

- [1] MPEG License Association: <http://www.mpegla.com>
- [2] MainConcept: <http://www.mainconcept.com>
- [3] Real: <http://www.real.com>
- [4] Darwin Streaming Server: <http://developer.apple.com/darwin/projects/streaming>
- [5] Flumotion: <http://www.flumotion.net>
- [6] Andy Wingo, "So you want to stream a conference": <http://wingolog.org/archives/2006/07/07/so-you-want-to-stream-a-conference>
- [7] Xiph: <http://www.xiph.org>
- [8] Theora Directshow codec: <http://www.illiminable.com/ogg>
- [9] Theora QuickTime components: <http://xiph.org/quicktime>
- [10] Icecast: <http://www.icecast.org>
- [11] `ffmpeg2theora`: <http://www.v2v.cc/~j/ffmpeg2theora>

**Table 1: Features**

	Darwin	Flumotion	Icecast2	Real
Codec free?	Open, but patented	yes	yes	no
Firewire/DV	no	yes	yes	no
V4L	yes	yes	possible	yes
RTP/RTSP	yes	no	no	yes
HTTP	yes	yes	yes	yes

## THE AUTHOR

For several years Oliver was a sysop and programmer at Ars Electronica Center in Linz/Austria.



After finishing his studies in Philosophy, Linguistics and Computer Science he became an editor for the Bavarian Broadcasting Corporation. Today he is head of the Editorial Competence Center for Software and Programming at Linux New Media AG.