Integrate web video with MythTV

DVR MEETS WEB

With MythTV and a couple of plugins, you can merge Internet-delivered video right into your DVR playlist. **BY NATHAN WILLIS**

Web-delivered video has come a long way in a few short years – and I don't mean user-developed, self-published content, either. Major television networks now offer a substantial portion of their programming online, enough that it can supplement the traditional broadcast media (cable, over-the-air, and satellite). But, although these content sources are effectively video-on-demand, they typically are not available in a way that lets you easily integrate them into a broadcast-centric DVR system, like MythTV [1].

As always, though, the open source community comes to the rescue. I'll take a look at the different ways you can seamlessly merge online video sources with your MythTV system. With some plugins and helper applications, you can merge Internet-delivered video right into your playlist, regardless of whether you receive it via a BitTorrent, RSS, or Atom feed.

To be fair, I should note that some Linux "media center" applications make web video a priority, like XBMC and Boxee. If *most* of the programming you need to watch is available through feeds or websites, you might find it easier to install one of these applications and run merrily along. For the average user, however, no other open source media center application offers MythTV's robust support for delivering live TV, scheduling recordings, and managing recurring programs and subscriptions. For content delivered via live broadcast, it is a necessity.

MythVideo

In previous generations of MythTV, all video content that was not recorded from hardware tuners was accessible only through a separate plugin, MythVideo [2]. MythVideo could play back almost any media type, but the integration it provided with the rest of the system was poor. It used a separate storage directory and a separate database table, and it was buried in a separate part of the menu structure – as was its complement of settings and preferences. It even used a different interface for navigating between videos, more akin to a simple filesystem browser than MythTV's user-friendly Recordings screen. If you wanted to associate metadata with your files (i.e., "cover" images, dates, or synopses), you had to add that for each individual file.

That said, one good thing that MythVideo brought to the party was support for any and every media type: It didn't matter where the video came from, how it got to your computer, or whether it was from a series, a standalone work, or even something you shot with your phone. MythTV's Recordings interface is built around items that can be associated with an "air time" and a source channel. As I'll describe, several of the web video solutions find ways to cope with that set of constraints, with varying degrees of success.

Besides handling the lack of a "channel" for web video, the list of requirements for web video integration should probably include merging the new content into the same media browser as your television recordings as much as possible, so you can sort through all of your video in one place, search it, and even delete it when you're done. Although the ability to add or subscribe to new content from within MythTV would be nice, it is not absolutely essential.

MiroBridge

Next, I'll look at the most predictable of all video sources – those that come through a standard RSS or Atom feed. This list includes video podcasts and quite a few Internet-only programs, but not just indie productions or niche top-

ics. Some popular outlets, like The Onion, make a standard feed-based video source available. These feeds are a big win for MythTV users because they serve up the same content regardless of the application used. The broadcaster might be targeting iPad owners, but the standardized delivery mechanism makes that assumption irrelevant.

Currently, the nicest option for feed sources is a plugin named MiroBridge [3]. MiroBridge uses the cross-platform video podcast client Miro [4] to manage feeds and download videos in the background, but it also implements a "virtual recording device" to MythTV. The core functionality comes in a Python script named *mirobridge.py* that quietly runs the Miro app to check your subscribed feeds for new content in a background process, downloads any new videos available, then shuts down Miro and inserts the new videos into MythTV. The result is that subscribed videos appear in the Watch Recordings list alongside normal television recordings, complete with the appropriate program and episode names. They are marked as originating on the Miro Channel, which, of course, does not appear as a live TV source.

Many MythTV distributions (including Mythbuntu) now package MiroBridge as a standard option (Figure 1). Mythbuntu even provides a configuration tool in its Mythbuntu Control Centre, but if you installed your own MythTV package or don't use Mythbuntu for other reasons, don't worry. Most of the effort involves configuring Miro's storage and feed preferences to be compatible with MythTV's expectations.

At the time of writing, the current release requires MythTV version 0.22+ fixes or later, but you should check the project's homepage before proceeding. You will need to install Miro itself on your MythTV back end (remember, Miro acts as a recording device, so it must be accessible to the *back* end). Any recent version should suffice. Also, you will need Python 2.5 or greater, ImageMagick, FFmpeg, and the pyparsing Python library. All of these components should be available through the package manager.

Next, I'll walk through the setup process. Mythbuntu Control Centre abbreviates several of the steps into a series of buttons and drop-down selection menus, but you should understand what they do. To begin, configure MythTV's storage directories for graphics. In the MythTV front end, open *Utilities/Setup* | *Setup* | *Media Settings* | *Video Settings* | *General Settings*. You can choose directories for movie posters, screenshots, banners, and fan art. Whichever directories you choose, make sure they will also be accessible to Miro.

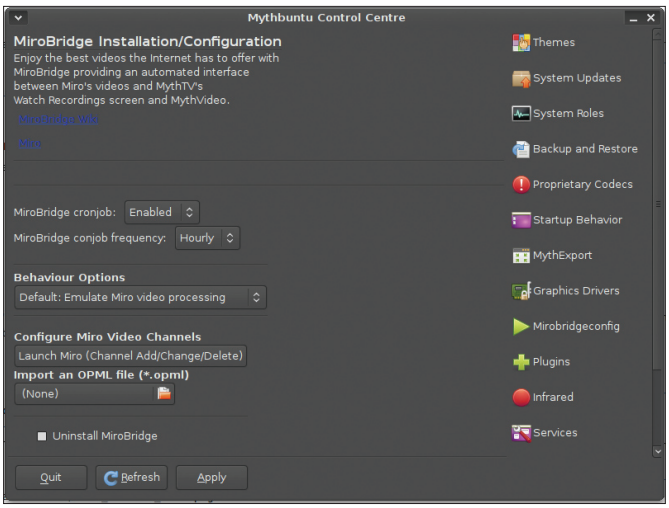

Figure 1: Mythbuntu's MiroBridge setup tool simplifies the installation and configuration process.

Next, you will want a "Miro Channel" icon – Mythbuntu Control Centre installs one for you automatically, but you can also grab it from the MiroBridge site and save it somewhere on your MythTV back end. If you use channel icons for your other channels, the easiest solution is to save the Miro icon in the same directory. Then, run the MiroBridge script from a command prompt, passing it the location of the icon with the *-C* flag. For example:

```
/usr/share/mythtv/contrib/imports/
    mirobridge/mirobridge.py
  -C /home/nate/icons/
    miro_channel_icon.png
```

Because the path to *mirobridge.py* might vary a bit depending on your distro, run *locate mirobridge.py* to find yours.

By default, MiroBridge assigns the virtual Miro Channel to channel number 999. If you have a *real* channel 999 in your lineup, you can change this setting with the *mirobridge.py* script. To assign Miro to channel 12345, run:

```
/usr/share/mythtv/contrib/imports/
 mirobridge/mirobridge.py -c 9999:12345
```

In this example, the *9999:* corresponds to the internal XML channel ID, which you probably do not want to mess with, but the *-c* flag requires it.

Now, you can launch Miro and find some feeds. You can browse the Miro project's *Miro Guide* directory, which is filled with popular and recommended sources, and subscribe to those you want to record regularly (Figure 2). Be sure to remove any unwanted feeds that come pre-selected in the app's Video Feeds list, though – it can be overkill.

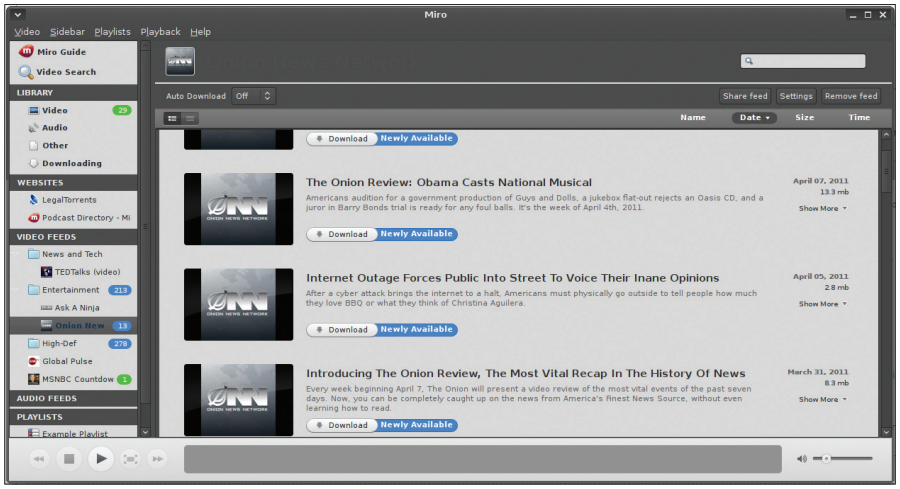

Figure 2: Subscribing to video feeds in the Miro application.

When you're finished adding and removing subscriptions, right-click on any that include non-alphabetic characters or have a very long feed name; then, choose *Rename*. You want to rename these feeds to a shorter title without extra parentheses or punctuation marks, because they can confuse MythTV's media browser. When you are done, quit Miro.

You also get to select how MiroBridge treats videos. The default is to keep unwatched videos in the Recordings group and move watched videos into MythVideo (hence the need to set MythVideo storage directories for graphics above). I strongly recommend changing this to the behavior that treats all Miro videos like normal Recordings, which expire according to the normal MythTV rules.

In Mythbuntu Control Centre, choose *Watched Recordings screen only* under *Behaviour Options* and click *Apply*. For a bit more control, you could also copy the *mirobridge-example.conf* configuration file from the MiroBridge installation directory to your own *.mythtv* folder, renaming it *mirobridge.conf*. Here you can configure MiroBridge's behavior for watched videos. You can make the same behavior change by moving the line *all miro channels =* from underneath the *[watch_then_copy]* section to underneath *[watch only]*.

MiroBridge is intended to run periodically as a cron job – daily, hourly, weekly; the choice is yours. Here again, Mythbuntu Control Centre gives you a selection box with Hourly, Daily, and Weekly options. For more control, you can start the crontab editor with *crontab -e* and add a cron entry manually.

The following daily cron slot:

```
45 02 * * * /usr/share/mythtv/
  contrib/imports/mirobridge/
  mirobridge.py
 -V "/tmp/mirobridge.log" 2>&1
```

will run the *mirobridge.py* script at 2:45am every day, logging the output to */tmp/mirobridge.log*.

MiroBridge is a bit of a hassle to set up, certainly, but a large part of that stems from its reliance on an external application (Miro itself). Once MiroBridge is up and running, however, all of your feed-delivered video will be automagically downloaded just like recorded television programs.

MythMediaMonitor

MiroBridge limitations include the requirement that Miro also be installed on the back-end machine. If your MythTV back end is a headless server that pulls in all sorts of other GUI package requirements, it can be difficult to configure Miro. But, the main shortcoming of MiroBridge is that it only handles Miro's feed-delivered videos, not BitTorrent or other manual downloads. To get these other forms of video into MythTV, however, you can use MythMediaMonitor.

MythMediaMonitor [5] is a Bash script that, when run periodically, looks for new video content in any directory you specify. When it finds something, the script is smart enough to parse the file name, determine the title and program number (if there is one), and add the new content to MythTV's Recordings (Figure 3). Thus, in practice, you can set MythMediaMonitor to watch a *Downloads* directory and, if your BitTorrent client supports moving completed files to it, your completed downloads will automatically appear in your MythTV Recordings list.

Also, if you own a movie or a series on DVD and want to rip the video into MythTV for easy viewing, you can have MythMediaMonitor watch that directory as well. Certainly some people will prefer the separate "archival" features of MythVideo for that sort of long-term storage, but having the option is nice.

Compared with MiroBridge, MythMediaMonitor setup is a snap (Figure 4). No out-of-the-ordinary dependencies are required – in fact, apart from ImageMagick and the (optional) mkvinfo and mkvextract tools needed for Matroska file support, the only dependencies are standard GNU utilities like sed, grep, and sleep. Simply download and save the script on your MythTV back-end box (*/usr/local/bin/* is a safe choice) and open it in a text editor.

In the editor, scroll down to the *Options* section to find a well-commented list of variables that you'll need to change to fit your exact setup – mostly common details like your MySQL username, database name, and database password. You can also specify separate directories for TV episodes and movies, and you can also tell MythMediaMonitor not to add movie content to the Recordings list. As with MiroBridge, the default directory choices are the same as those used by MythVideo, but you don't have to keep them that way.

You will need to specify the MythTV recording group to use; the default is *Default*, which is fine for most people. And, you'll need to specify a Channel ID number. MythMediaMonitor wisely defaults to *2111*, so as not to conflict with MiroBridge, but you can change it if you experience a problem. MythMediaMonitor can also extract subtitle tracks from Matroska videos and convert them to MythTV-compatible SRT format; to do so, you must specify the subtitle language you are interested in. Finally, if you set *DEBUG = true*, the script will log events to MythTV's event database, which can be useful if you need to track down problems.

Other than that, just set a MythMediaMonitor cron job in crontab for as often

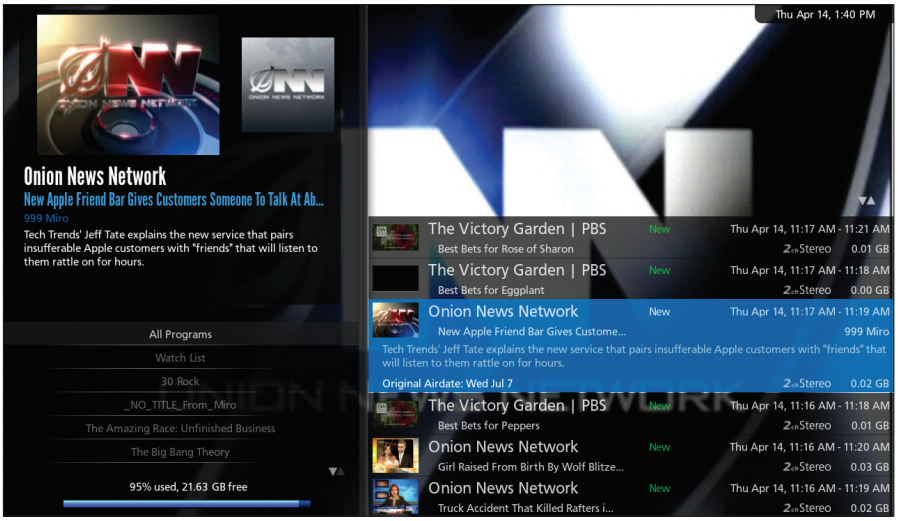

Figure 3: MythTV's Recordings screen, with web video inserted courtesy of MiroBridge.

as you'd like to re-scan your watched directories. As an alternative to the daily MiroBridge example earlier

```
23 * * * * /usr/local/bin/
  mythmediamonitor.sh
```

will run the script once every hour, at the 23-minute mark. Once a video appears in one of the watched directories, the script will fetch all of the relevant information it can retrieve from the web (currently using IMDb and TVRage.com), including titles, episode numbers, synopses, original air dates, and thumbnail images. Once an item is in the Recordings group, your MythTV expiration rules determine when the video gets deleted.

Of course, like MiroBridge, MythMediaMonitor requires an external application to find the video you are interested in: a browser, a BitTorrent client, or a disc-ripping utility. But unlike MiroBridge, there is no "set it and forget it" analogue to subscribing to a feed. MythMediaMonitor is built for videos that you find and fetch on a case-by-case basis. That use case can't really be integrated into MythTV itself (unless you count starting a download from MythBrowser).

Interestingly, Miro actually has BitTorrent downloading capabilities built-in, but they are not exploited by the MiroBridge plugin. So, you could use the two together, perhaps approximating a more seamless media-fetching experience.

MythNetvision

RSS feeds and BitTorrent or HTTP downloads are fine and good, but they have one fatal flaw: Many vendors in the broadcast industry skip them entirely and instead build their own custom, non-standard, Flash-based and quirk-filled web delivery platforms. Most of these broadcaster-run web video services function only in the browser, and they have little in common around which to build a consistent interface. What's a MythTV user to do?

The answer from the MythTV project is the new plugin MythNetvision [6], which is poised to bring full integration of online video into the MythTV experience, including custom broadcaster sites, RSS and Atom feeds, and browser-only video, with support for searching content online and watching it directly within MythTV.

Not all of the pieces are in place yet, but there's solid support for a healthy set of web video sites, including video sharing sites like YouTube, Blip, and Dailymotion, plus television broadcasters like Comedy Central, BBC, PBS, MTV, and even the multi-network Hulu service. MythNetvision supports individual video sites by separating functionality into a core plugin plus a set of site-specific "grabber" scripts. A well-maintained set of grabbers ships with the plugin, but users can add their own.

The MythNetvision plugin is provided by all major MythTV distributors and Linux distros, so installing it is a matter of selecting the appropriate package from your package manager. When installed, however, you will have configuration options that you need to choose through your MythTV front end.

The plugin adds two menu items to the Media Library screen: *Browse Internet Video* and *Search Internet Video* (Figure 5). In the Browse screen, you can navigate through the supported video sites just as you would recordings; in the Search screen, you can perform keyword searches and browse through the results. Search comes pre-configured with every grabber available as a selection; Browse does not. Visit the Browse screen, then press the menu key (keyboard *M*). The pop-up menu options are: *Update Site Maps*, *Update RSS*, *Manage Site Subscriptions*, and *Manage RSS Subscriptions*.

Your first time through, you must choose *Manage Site Subscriptions*, which lists all installed grabbers. Check (or uncheck) them to select the sites you want to browse (Figure 6). Then, you can hit Esc and return to the previous menu. Choose *Update Site Maps* and MythNetvision scans the sites and assemble the content for your browsing pleasure (Figure 7). This process can take quite a while, depending on how many "site subscriptions" you select.

The same process is required to use the plugin's RSS subscription functionality, but I don't bother with RSS in MythNetvision because it duplicates the functionality of MiroBridge in a less convenient way. MythNetvision already segregates its web video browsing from Recordings; further separating RSS and Site grabbers makes no sense to me.

In daily use, MythNetvision's support for web video isn't the full integration I would like, but it is the best option available for broadcasters that insist on isolating their own content streams. As for the supported grabbers, only about 20 are currently available, but they are a well-curated collection. The MythTV team insists on providing grabbers that adhere to the sites' Terms of Service; if you want to skirt the TOS, you can find

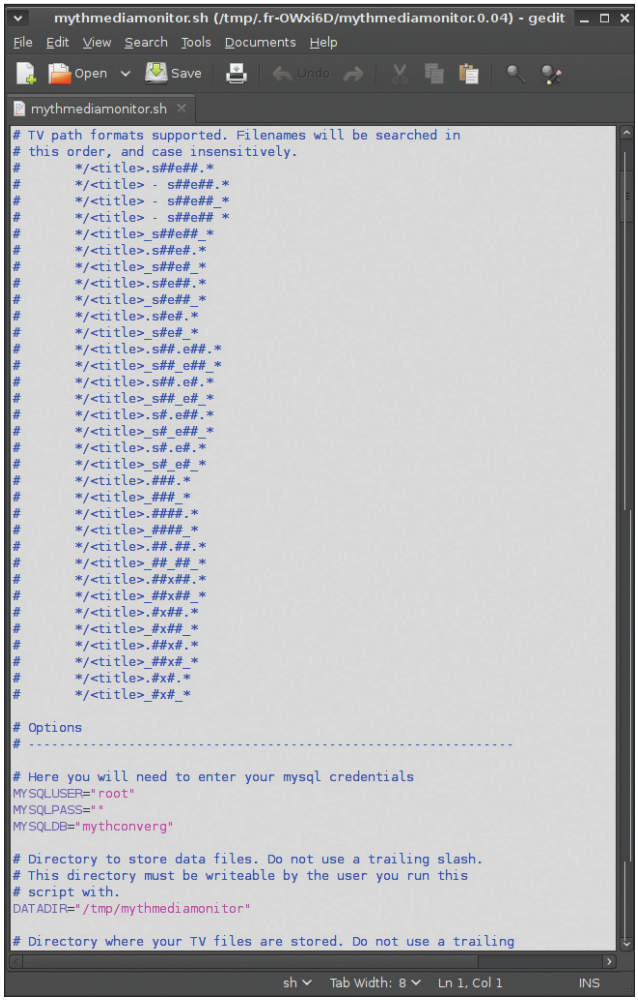

Figure 4: Editing the MythMediaMonitor configuration script.

independently developed grabbers for other sources, often by searching users' personal pages on the MythTV wiki.

The wiki is also the best place to look for unstable work-in-progress grabbers, but be forewarned that some of the newest grabber code only works with the newest MythTV builds. The fun part is that you can study all of the grabbers online and maybe even write your own.

Figure 5: The MythNetvision plugin adds two new features to the Media Library.

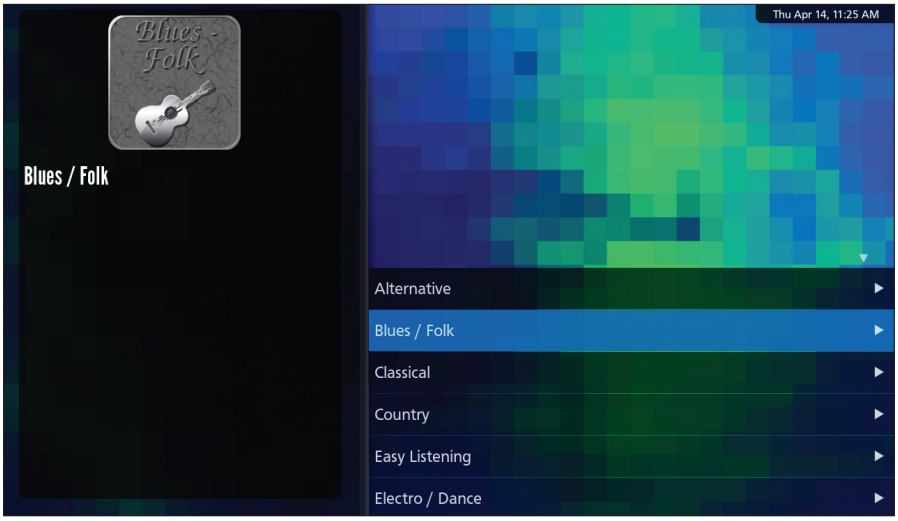

Figure 6: MythNetvision grabbers implement a simple menu-based browser for video content. Here you see music performances provided by MTV.

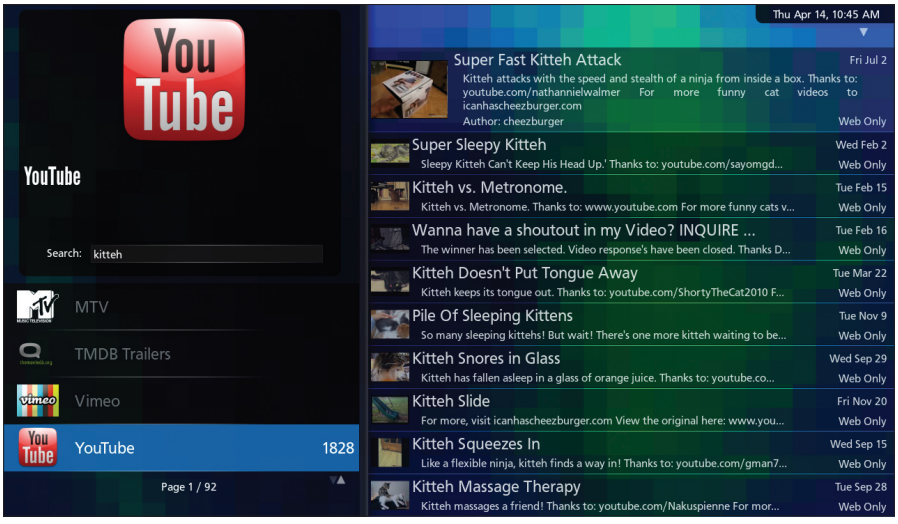

Figure 7: MythNetvision can also send search queries to any (or all) of your subscribed grabbers.

Moving forward, MythNetvision still has a lot of room to grow. For one thing, I find the terminology in the interface obtuse: "Update Site Maps" sounds like it has nothing to do with web video at all. More importantly, I am hopeful that, in the future, it will adapt to fit more of MiroBridge's integrate-feeds-with-recordings model, which genuinely makes more sense. After all, RSS subscriptions should be just another "recording device," shouldn't they? It's all just video.

Closing Credits

Full web/broadcast integration isn't quite here yet, but it may never arrive: Standards change too fast. For example, I'm not sure whether MythTV developers mean to support Google's open source WebM codec, which would improve quality and reduce bandwidth for web video.

Still, putting together the three pieces I've discussed here effectively makes it possible to use MythTV for all of your viewing needs. With MiroBridge, the RSS feed is considerably more reliable, and using it frees up a hardware tuner.

On the other hand, it's impossible to forget that web-delivered video is still a different beast from recordings you grab with a DVR. You're not in control with web video; the site can be slow to respond, use a clunky UI, suffer from buffering problems, or only offer a small subset of a particular show's episodes. Some of that might change with time as broadcasters warm up to the idea that eyes are eyes – regardless of the delivery medium. ■

INFO

[1] MythTV: *http://www.mythtv.org/*
[2] MythVideo: *http://www.mythtv.org/wiki/MythVideo*
[3] MiroBridge: *http://www.mythtv.org/wiki/MiroBridge*
[4] Miro: *http://www.getmiro.com*
[5] MythMediaMonitor: *http://www.nowsci.com/mythmediamonitor/*
[6] MythNetvision: *http://www.mythtv.org/wiki/MythNetvision*

AUTHOR
Nathan Willis is a freelance journalist who writes about Linux and open source.