

BitTorrent: Set

Neil Bothwick explains how BitTorrent works and how you can set up your own file and directory sharing, and even your own torrent tracker.



Our expert

Neil Bothwick has a computer in every room, but won't disclose the location of his central server for security reasons.

Most Linux users have used BitTorrent at some time, at least those of us with a reasonable internet connection. In contrast with many groups, we see it as a way of sharing files that should be shared, not as a means of downloading content that we shouldn't. But how many of us have given it more than a passing thought? How does it work? How can we use it to share our own files? Can it be done privately, without making our files available to every user of The Pirate Bay?

Rather than downloading a file from a single-source server the BitTorrent protocol enables several users to join a 'swarm' of hosts to download and upload from each other at the same time, sharing the bandwidth burden between them. With BitTorrent there is no concept of a server containing the file and clients downloading it; every computer running a torrent is doing both. You don't even need the whole of the file to be in one piece in any place, although that helps. BitTorrent splits a file into pieces, and the torrent describes the details of this for a particular file or collection of files. When you start downloading from a torrent, your BitTorrent program will begin downloading pieces from those who are sharing them. These are not necessarily from the same place, nor in any particular order. Once you have downloaded complete pieces, they are made available for sharing, so you are sharing the file almost as soon as you have started downloading it.

BitTorrent refers to seeders and leechers. Seeders are computers that have downloaded the complete file but continue to share it. It's good manners to leave the torrent running after the download has completed to allow others to share it. Leechers are those who've not yet completed the download, but this is a rather unfortunate term as leechers are seeding as well, as everyone is an uploader with BitTorrent.

So how does your BitTorrent program know where to get the pieces of the file you need? This is handled by trackers. These are computers that keep details of torrents, they act as a sort of clearing house, knowing who has which parts. The address of one or more trackers is embedded in the torrent file, so your client knows where to start looking. The trackers

are not involved in the transfer of the files themselves, they simply let you know where the pieces are.

Do not confuse a tracker with an indexer. An indexer is a website that lists details of torrents and lets you download them, like The Pirate Bay (which hosts a lot of legal content, despite its name). Some indexers also run trackers, but this is a convenience, not a necessity.

There are plenty of clients available; my personal favourite is *Transmission*, which also happens to be the default in Ubuntu. If you want something to be used without a GUI, *Transmission* has a terminal client and a headless daemon, or you could try *rtorrent* which is also very popular.

Sharing your own files

So that's how BitTorrent works in a nutshell, and you've most likely used it at some time to get hold of a distro's ISO image, for example. But you can use it to share your own files too. There are a number of reasons why you may want to do this. The most obvious is bandwidth. Unless you have a server with a fast internet connection, serving your new distro respin over your DSL or cable internet connection is going to be painfully slow, especially if it becomes popular. The more people that want it, the slower it gets. By contrast, torrents become faster with popularity, because more are sharing them.

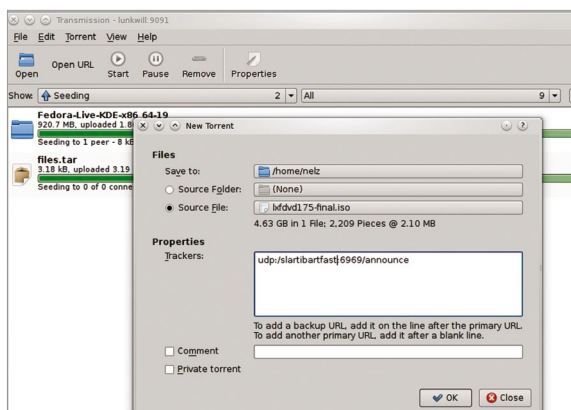
Another reason is speed. You could upload your files to a fast server, a VPS or download site, but they would have to be completely uploaded before anyone could start downloading. With BitTorrent, the files are available immediately and, as soon as people start downloading them, at a speed greater than your own upload rate.

"But isn't BitTorrent public?" I can hear. No, not really. The trackers simply track who has which pieces of which torrent, but knows nothing about the torrents themselves except for the hash key used to identify the torrent. The files cannot be downloaded or shared without the torrent file, and it's up to you how you share that – whether that's with the whole world or just your circle of friends, BitTorrent is suitable.

The key to all of this is the torrent file: how do you create one? There are several tools for this, and most of the desktop BitTorrent clients have an option to do so. For example, in *Transmission*, select File > New, pick the source (which can be either a directory or a single file) and type in the URLs of the tracker(s) to use. How do you know which trackers to use? There are plenty to choose from, such as PublicBitTorrent (<http://publicbt.com>) or alternatively OpenBitTorrent (<http://openbittorrent.com>), the websites give the tracker address to use. It's worth adding more than one tracker, so if one is down or overloaded, your torrent will still work.

There are also command line programs that can be used to create torrents when a GUI is not available, or from a script. *Mktorrent* and *createtorrent* are two choices, but I always end up back with *Transmission*:

```
transmission-create -t udp://tracker.publicbt.com:80 -t udp://
```



➤ Most BitTorrent clients have an option to create a torrent file with a couple of mouse clicks, provided that you have a tracker address.

up a torrent

`tracker.openbittorrent.com:80/announce somefile`

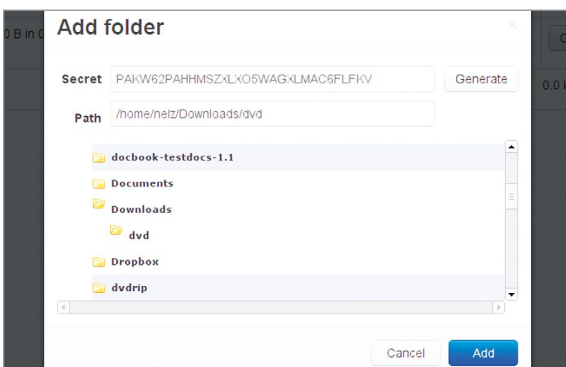
will create **somefile.torrent** – where 'somefile' can be a file or directory, which you can share with whoever you want. But where do they download the content from? At least one person, usually you, has to seed the torrent. Copy the file to share (not the torrent) to the directory in which your BitTorrent client saves completed downloads, then load the torrent into your client. Your client will see that the file is already there and, after verifying it against the checksum included in the torrent, will start seeding it. You need to keep seeding the file until at least one copy of each piece has been downloaded by others. At this point you could stop seeding and the file would still be downloadable even if no single user had the whole file, but you generally want to keep seeding your own files if you want to be sure they are always available.

Track yourself

There's something about the word 'tracker' that sets off the privacy alarms, so even though a torrent tracker does not look at the contents of the files you are torrenting, you only have the word of the tracker operator for that. If you are really concerned about privacy, the answer is to only use a tracker operator you really trust, and who do you trust more than yourself? That's right, you can run your own torrent tracker. You could even run a tracker that is only visible on a private network for local(ish) sharing of files only. The most popular option is *Opentracker*.

Unusually, for such a well used piece of software, there are no packages available for it and not even a source code tarball – you'll need to get it from the CVS server. Provided you have a compiler environment, usually provided by the **build-essential** package, and the **cvs** package installed, you can build *Opentracker* by creating an empty work directory, **cd**'ing into it and running these commands, as your normal user:

```
cvs -d :pserver:cvs@cvs.fefe.de:/cvs -z9 co libowfat
cd libowfat
make
cd ..
cvs -d:pserver:anoncvs@cvs.erdgeist.org:/home/cvsroot
co opentrackercd opentracker
```



» **BitTorrent Sync** is another way of sharing files, although more suited to folder synchronisation.

make

This builds two executable files, **opentracker** and **opentracker-debug**, which you can copy to somewhere in your path, normally either **/usr/local/bin** or **~/bin**. There are no other dependencies to build or install but it does help to have a config file, to avoid specifying all options on the command line, there is a sample included. Then start the tracker with:

```
opentracker -f /path/to/configfile
```

This must be on a static IP address that's accessible for the clients wanting to use it, which includes forwarding the correct port from your router if you want access over the internet. Then you'll need to add this tracker to your torrent files when you create them. With the default *Opentracker* configuration, this would be:

```
udp://IP-ADDRESS:6969/announce
```

using the appropriate IP address or host name for the opentracker server.

There is also a Python script called **bttrack.py** included with the *bittornado* package. While this can usually be installed from your distro's package manager without the need for CVS or compiling, it is a little more fiddly to use, but you may prefer to try it this way. **LXF**

» **Public BitTorrent** is one of several services that provide pure tracking services, without trying to index all the torrents on the internet. Or you could run your own with the same software.

BitTorrent Sync

There is another option for sharing files, or even directories, using the BitTorrent protocol without torrents. The new BitTorrent Sync program (<http://labs.bittorrent.com/experiments/sync.html>) syncs directories on different computers, rather like Dropbox but without storing the files on a server. It is

easy to use, although appears rather slow for large files, and can be used to share multiple directories, with different users if you want. It's not open source and it does communicate with their servers, so it's not entirely private. Your choice on whether to use it depends on how paranoid you are.