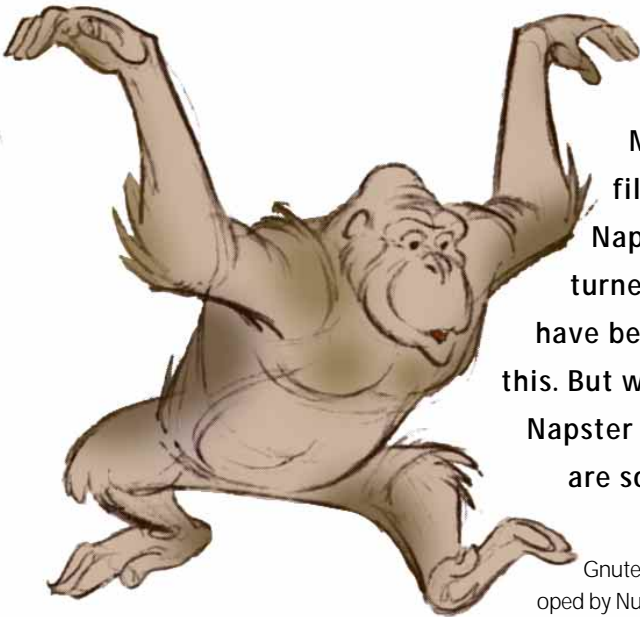


Distributed file sharing SUCCESSOR TO NAPSTER

BJÖRN GANSLANDT



Recently there has been a lot of uproar about Napster, a program that allows users to exchange MP3 files and, with the aid of Wrapster, other types of file as well. At the end of July a US judge ruled that Napster should be shut down, a decision that was overturned on appeal a few days later. No doubt there will have been further twists in the story by the time you read this. But whatever the final outcome, the fact remains that Napster isn't the only program or service of its kind. There are some interesting successors. Björn Ganslandt tunes in.

Gnutella was originally developed by Nullsoft which has now been part of AOL for some time, and when AOL merged with Time Warner the development of Gnutella was officially stopped. Since Time Warner sells CDs, the reason behind this decision seems fairly clear. However, despite this there are betas of the Windows program to be found online, not to mention many clones of Gnutella. Some of these are for Linux.

What distinguishes Gnutella from Napster is that there is no central host via which everything is handled. Instead, programs pass search requests to each other to find out who has the required file. Thus all Gnutella programs are both server and client. This approach means that the system continues to exist even if a program fails. Interesting files are quickly acquired by many different users providing many points from which they can be downloaded by others.

The disadvantage of this system is that you need the address of another Gnutella user in order to take part in the network. Such an address can be obtained from <http://gnutella.nerdherd.net/cgi-bin/hosts.cgi>. A further problem is that Gnutella is not completely anonymous. It is possible with no great difficulty to log the IP addresses of all users who download a file from the local Gnutella. Gnutella was originally designed by Nullsoft for use by relatively small groups; therefore it is rather slow in the opinion of the many users.

A good Gnutella implementation for Linux (and Win32 too) is Gnut. Gnut is relatively mature but to date still has no stable graphical interface. The "help" command displays a list of available commands. Gnut is configured using the file `~/gnutrc`. In the file `~/gnut_hosts` the addresses of all known hosts are stored. Gnut can display all search inquiries received using the command "monitor". When searching with Gnut and all other Gnutella programs you must avoid using wildcards (*, ?) as they are not recognised by all implementations. It's best to search only for keywords. Gnut can be downloaded from <http://www.umn.edu/~jlp>.

A nicer program to use is Gtk-gnutella. It emulates the look and feel of the original Gnutella and runs with no problems. The interface is fairly intuitive: IP addresses are entered in the format "Address:Port" in the "Add" field. Both binaries and source code for Gtk-gnutella can be obtained from <http://gtk-gnutella.sourceforge.net>. As the name suggests, you need GTK+ for Gtk-gnutella to run.

Another Gnutella implementation having a graphical interface is Gnubile. With Gnubile you must enter all IP addresses using the menu *File, Preferences* in the format "Address Port". Apart from this the program is also self-explanatory. A requirement for using Gnubile is GNOME. The program can be obtained from <http://gnutella.wego.com>.



LinuxMagazin/gnutella

If you would only like to sniff around in Gnutella there are some Gnutella search engines on the web. For a start, try the addresses <http://www.surfy.com/page/nut.php3> and www.dailyphat.com. It's also worth visiting <http://gnutella.wego.com> if you're looking for more clients or for information on Gnutella.

Jungle Monkey

Jungle Monkey is relatively new: consequently, hardly any files can be found with it. Despite this, the system is interesting. Jungle Monkey provides several channels in which files can be offered. Each user can open these channels, exchange files and chat with other users.

Jungle Monkey differs from other programs like Gnutella in that it uses **multicasting** (*end-to-end multicasting* to be precise).

First, Jungle Monkey opens a "root channel" which carries information about all the other channels. As you cannot get information about the other channels without this root channel, Jungle Monkey has the same weakness as a centralized system like Napster. It is possible to open channels without having the root channel, but only if you have a note of the addresses.

In the channels themselves you can then download files. However, it can be a while before all files are displayed. Information about which files are available in the channel is only sent about once every ten seconds. If no other root channel is selected Jungle Monkey opens the standard channel on start-up.

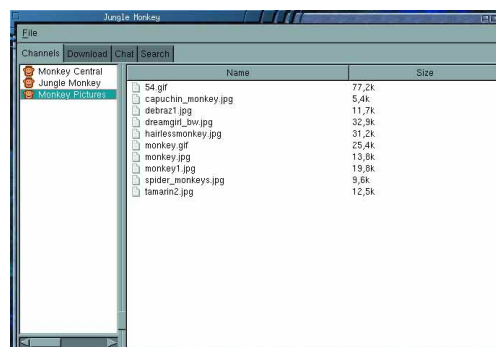
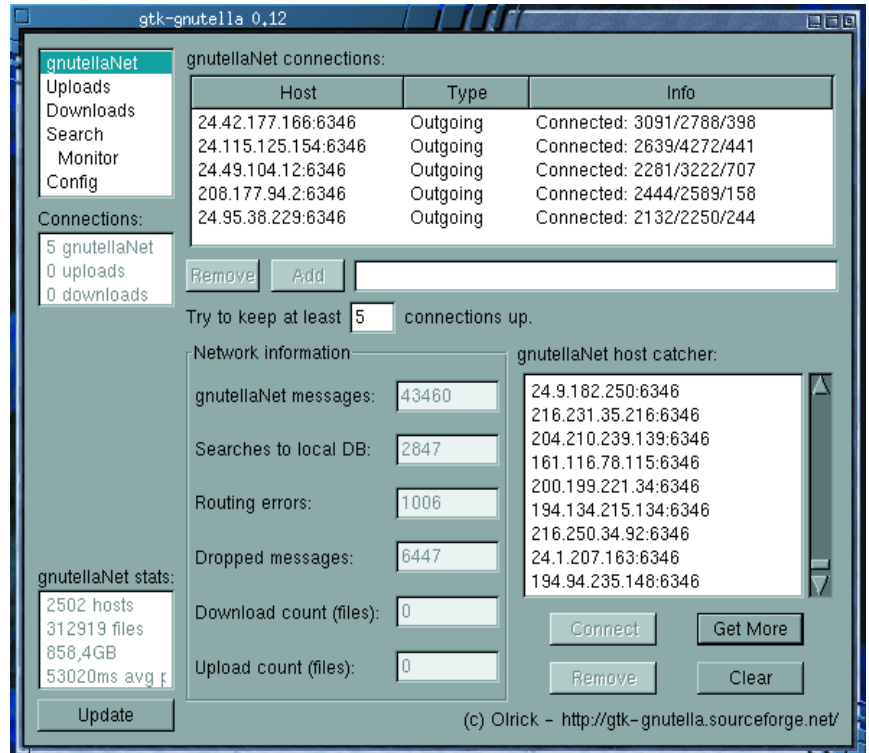
Jungle Monkey exists in both a GTK and in a GNOME version, the GNOME version being recommended. In addition, there is a command line version called *jmlite* and a command line search server named *jmsearchserver*. Information about Jungle Monkey as well as source code and binaries can be found at <http://www.junglemonkey.net>. (If this page cannot be found, try <http://141.213.11.223>.)

Freenet

Freenet is an attempt to create a network in which information can be anonymously procured and made available. Just like Gnutella, Freenet is a distributed and decentralized system. The system consists of a network of nodes. From these nodes you can download files by sending a key to a node. Keys have a format such as "text/philosophy/sun_tzu/art_of_war".

In this system it does not matter where a file is physically stored. If the first node does not have the file, it simply asks the next one. The whole thing functions in a similar way to Gnutella, except that Freenet has been designed for larger networks and the protocol is therefore rather more complicated.

The current version of Freenet still doesn't provide any anonymity or encryption but the next version should be relatively secure. At the moment there are still some problems with the architecture of Freenet. Thus, for example, it is still not possible



to update or efficiently search through documents. However, these problems should be eliminated in future versions. You can obtain more information on the protocol and the philosophy behind Freenet, as well as the program itself, from the Freenet home page <http://freenet.sourceforge.net>.

Freenet is written in Java, and is therefore readily portable to other platforms. However, there are also clients in Perl and C. The Perl client called "Liberator" is part of the Freenet software; FreeClient, the C Client, can be found at <http://lar2.dyndns.org> (this page is not constantly online.) You can also get limited access to Freenet via the Web (<http://uts.cc.utexas.edu/~blanu/keyindex.html>) although of course the access is then not anonymous. For the Java version of Freenet you will need a Java implementation installed on your system.

The future

There are some good implementations of Gnutella with which you can find a vast range of different files. However, for the future Freenet opens up some interesting possibilities. Take a look at it! ■

[above]
Figure 1:
Gtk-Gnutella

[left]
Figure 2:
Jungle Monkey

Multicasting: Using multicasting you can send a package of data from one source to many targets at the same time. This saves bandwidth compared to the more commonly used unicasting.
