davorr, Fotolia

Free communications on the Freenet network.

LIVING FREE

The Free Network Project provides a safe environment for free speech – even for users who fear censorship. **BY PETER CONRAD**

Freenet [1] is a network of computer nodes that use encrypted communications and anonymous file storage. The purpose of the Freenet network is to provide an anonymous environment for users operating in totalitarian countries who want to exercise their right to free speech without fear of government censorship. Freenet is also useful for corporate whistle-blowers or anyone else with strong feelings that might be too controversial to state openly.

Freenet is essentially a peer-to-peer network with many safeguards built in for preserving anonymity. Participants in the Freenet network operate as independent nodes. Each node only knows its nearest neighbors, and no node has complete knowledge of the network structure.

Users can upload files to the network and store them under a key. Others can use the key to download the files off the network. Anybody can download the file from any node on the network, even if the node that originally receives the request doesn't have a copy. If the node receiving the request does not have the file, it asks a neighbor, which might also need to ask its own neighbor, and eventually the request will reach every node on the network if necessary. All of this background communication is completely transparent to the user.

Freenet itself thus provides only an infrastructure that supports the secure, anonymous exchange of data. A collection of Freenet-ready client applications offer services such as web posting, file sharing, email, and message boards, and an http gateway lets users surf Freenet like the Web (Figure 1).

In all, Freenet plus its constellation of client applications support an environment that is like an anonymous, less interactive version of the Internet itself. Anyone interested in joining the Freenet network can do so by running a node. To do so, you have to install a Java application that implements the node, then you set up links to friends and acquaintances

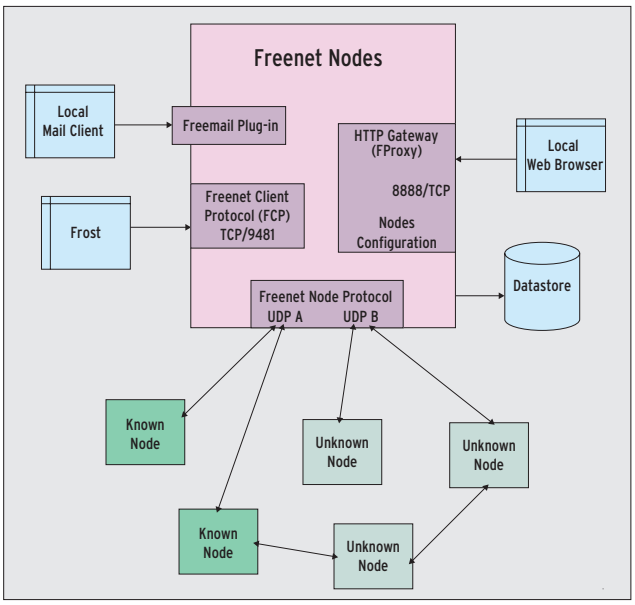

Figure 1: Freenet is a network of loosely connected computer nodes that use UDP for an encrypted data exchange. The implements basic functions for uploading and downloading data on Freenet. Clients use various interfaces to access these functions.

– or just let the machine search for other Freenet nodes. The Freenet Installer provides an address list with publicly accessible nodes.

First Steps

To install and run Freenet, you need a Java Runtime Environment – the newer the better. If you have installed Java WebStart, you can enable the installer by clicking on the download page. Alternatively, you can download, unpack, and run an installer in the normal way. The installer drops the Freenet node and various optional client applications into an arbitrary directory. Freenet just needs a couple of millibytes of disk space, but you will need at least 1GB for the data store – more space is better because additional storage increases the chance of finding a data package you are looking for locally.

The installer then looks for two free UDP ports and launches the node. The Freenet node – with its integrated http access feature – launches within a couple of seconds, and you can configure it using a convenient web interface (see Figure 3). If you are using Freenet from home and don't have a static IP address, you might want to use a dynamic DNS tool such as DynDNS to let others on the network associate the node with a static DNS name. The DNS name, the bandwidth restrictions, and the OpenNet mode are important settings. If you are behind a NAT router, you can forward the UDP ports configured on the router to the computer running the Freenet node. The router only needs to allow responses to requests sent by your own node, and most NAT routers do this automatically.

After launching, the node starts to contact other nodes. Two possible scenarios are:

- If you have enabled OpenNet mode, the node will contact the *seed nodes* supplied by the installer and let them know its address and communication key. The seed nodes respond with addresses and keys of other nodes to which your node can connect.
- If you have friends or acquaintances that also run Freenet nodes, you can exchange contact data with them manually. Use the web interface to discover your own node's address and communications key, which you can then pass on to your friends. After entering the addresses and keys in your own configuration, the nodes will start to exchange data.

For workable network integration with the Freenet network, you will need to contact 15 to 20 other nodes. If you do not have so many friends on Freenet, Opennet mode is your only option.

Freesites and Flogs

The front page of the Freenet web interface is also the starting point for one of the most important Freenet features – a list of major Freesites. An ordinary browser provides you with access to the Freenet sites. The FProxy HTTP Gateway integrated into the Freenet node forwards the browser request and converts it into a Freenet key search. Some patience is required at this point – depending on the popularity of the page, the

Specialized

Freenet strives to ensure that clients find the shortest path to data with a high degree of certainty. Because each node has a limited amount of storage capacity, it needs to decide which files to store and which to discard. It thus prefers files that are similar to those it already has, where similarity is measured by the key hashes and not by the file content. Other nodes learn what their neighbors specialize in and monitor the files that they receive from these neighbors. A node will send a request for a certain key to a neighbor from whom it has previously received the most similar keys (Figure 2). This mechanism works best when connections to neighbors are stable. In other words, nodes should run permanently in the background if possible.

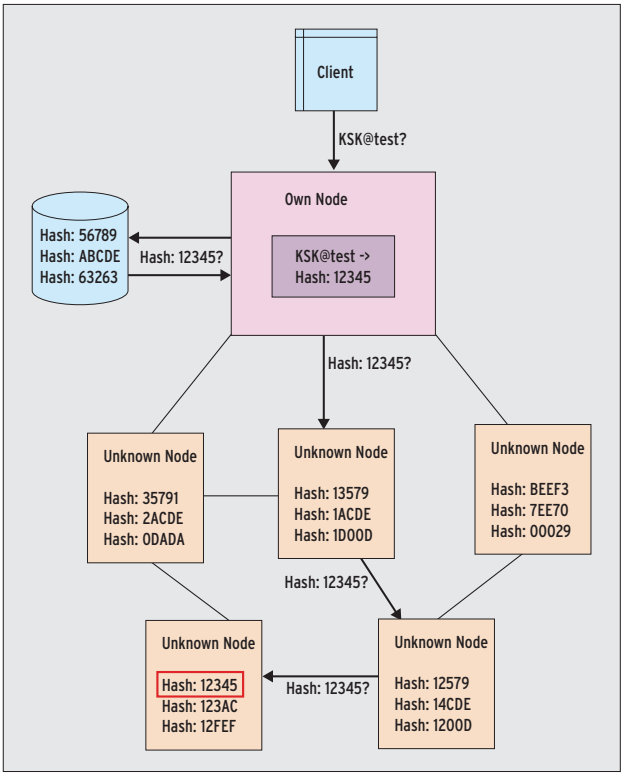

Figure 2: The node first converts a requested key into a hash. It looks for the hash in its own data memory before turning to connected nodes. The arrows show the request directions. After finding the file, it follows the same path back.

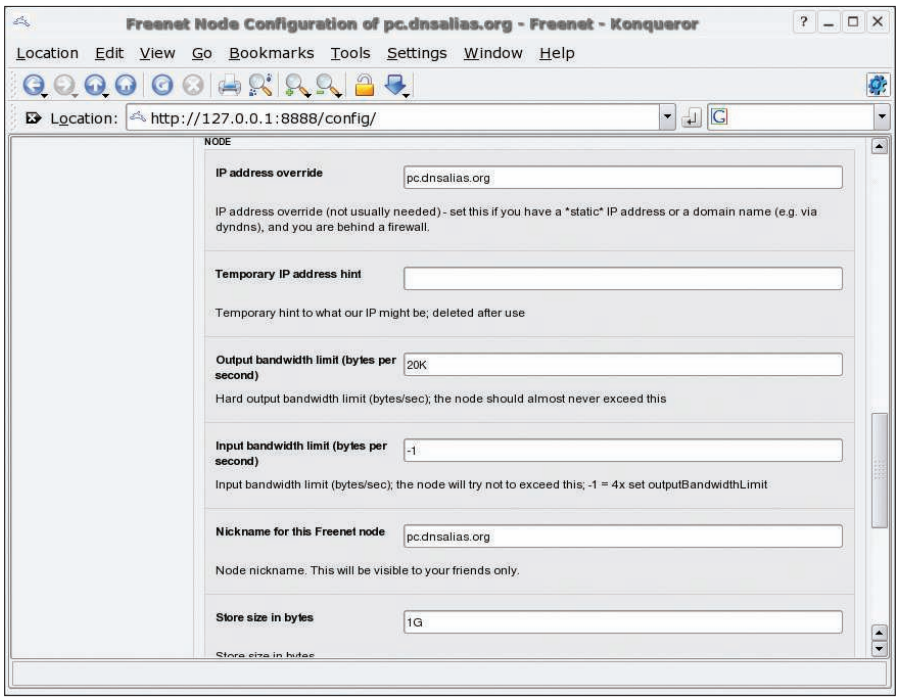

Figure 3: A web interface is used to configure the Freenet node. Besides the name, you can see the bandwidth and disk space assignments.

search could take a couple of minutes.

Freenet even has its own index pages. Freenet users have set themselves the task of categorizing existing Freesites and searching them for updates. Because Freesites do not offer the same kind of interactivity that a website offers, the index pages serve the role on Freenet that search engines serve on the web.

A flog is a Freesite equivalent of the blog. In addition to a number of Freenet developers, a large collection of colorful characters embrace the anonymity of Freenet by operating flogs.

Freenet 101

Freenet's simple tools offer a variety of options for creating and managing a Freenet site. To publish information, you can use HTML pages with style sheets and images, although active content such as JavaScript or Flash is filtered out for security reasons.

The graphical jSite tool, which you can install by running the *bin/install-jSite.sh* script, lets users upload finished web pages to Freenet. After completing the install, just click on *jSite.jar* or enter *java -jar jSite/jSite.jar* at the command line to launch the program. The dialog that appears can manage multiple projects. jSite first asks you for your own node's address. For a Freenet node running locally, you can accept the default of *localhost:9481* as the client port.

Selecting *Add project* in the menu tells jSite to create a new project and generate a keypair for the project (Figure 4). Users can add a path name and thus generate a USK. After assigning a name to the project and selecting the local directory that contains the files for your Freesite, you can then click *Next* to go to the site details. jSite lists all the files in the project directory. Clicking on a file name lets you specify how jSite should handle the file. The requirement is that you have at least an index page, such as *index.html*. jSite stores these settings in the project, which avoids the need to start from scratch whenever you update your project.

Containers are used to group files in ZIP format. Grouping multiple files in a container helps the page load faster. For occasional site updates, it is a good idea to group any files that change frequently in a separate container from the containers with more permanent files.

Clicking *Insert now* starts the upload. At this point, jSite packs the files into the containers and passes them to the Freenet node, which then forwards them to other nodes. Depending on the size of the site, this could take a couple of minutes. After completing this, your own Freesite is publicly accessible on Freenet: To transfer the Freenet address, you can select *Copy URI to Clipboard* on the jSite overview page, then you can cut and paste the address into the Freenet start page *Access a key* box.

Cooling the Trail with Frost

Frost is a Freenet tool that provides a collection of features such as newsreading and message boards [2]. The Frost utility comes in a ZIP archive. After unpacking in a separate directory, you can launch Frost at the command line by entering *sh frost.sh*.

The first time you launch Frost, it asks you for your Freenet version and a pseudonym. Frost links the pseudonym with its own public key, which others can uniquely identify even if a third party were to choose the same pseudonym. For example, a flog author can publish her pseudonym and key on her Freesite. Frost users can then be certain

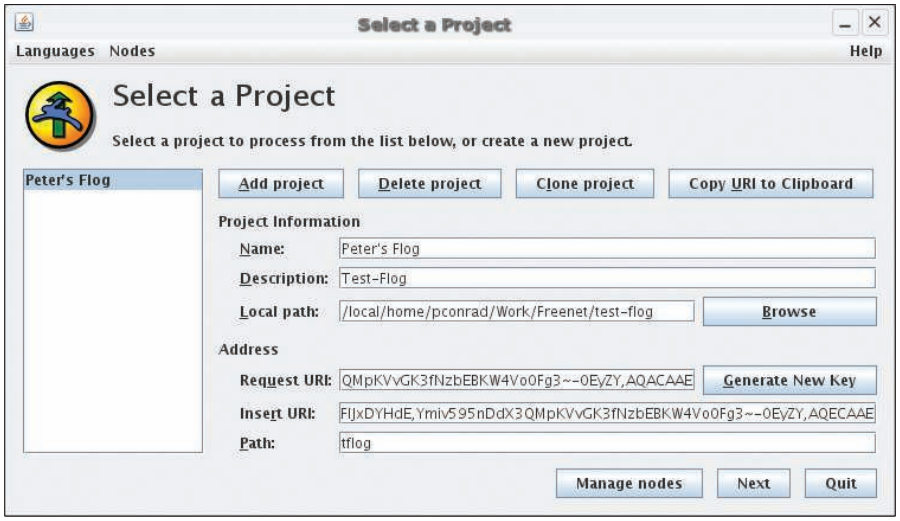

Figure 4: jSite lets users publish their own Freesites and flogs. A jSite project groups HTML files and matching graphics.

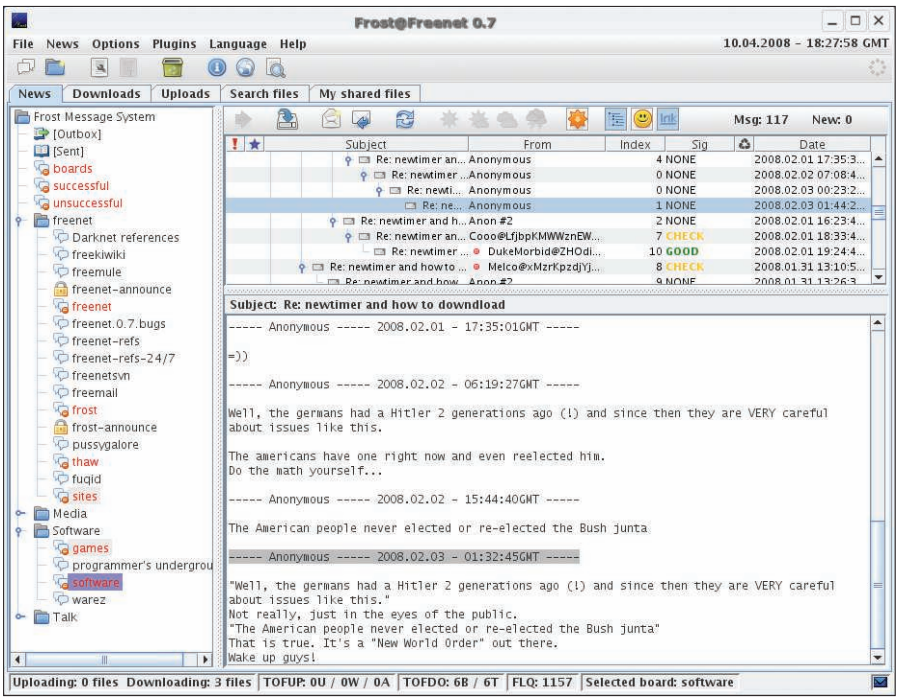

Figure 5: Frost offers similar mass communication features to Usenet but with pseudonym-based authentication. Participants can thus discuss controversial topics without disclosing their real names but can also be sure that their contributions are uniquely distinguishable and impossible to manipulate.

that the pseudonym belongs to the flog author. More pseudonyms can be created later as needed.

A user who wants to post a message can choose a pseudonym under which to publish – or opt to do without one. Frost displays a splash screen on launch that threatens to send your personal details to the secret service, which is attributable to the developer's warped sense of humor, but this dire warning is nothing to worry about.

Freenet Keys

Users or client programs use file keys to access files on Freenet. Freenet distinguishes between the different kinds of file keys that are suitable for various purposes.

Content Hash Keys (CHKs)

The *hash* is a fingerprint of the file content. If you want to request a document, you need *Document-Key* and *Options* to decrypt the file. Because the key is more or less uniquely linked to the file content via the fingerprint, it is practically impossible to modify the file content without the changes being noticed. CHKs are thus the backbone of Freenet.

Signed Subspace Keys (SSKs)

Like CHKs, *Document Keys* and *Options* are also used by the requesting entity to decrypt the file content, which is also digitally signed. The requester can use the *Public Key* to check the signature.

The SSK creator can assign any *Name* to differentiate between documents. Because nobody can change the content of a file posted on Freenet, not even the key originator, a *version number* has been introduced. The key originator can increment the version number and thus practically create a new key under which a modified (or completely new) file can be published.

Keyword Signed Keys (KSKs)

KSKs are the simplest and least secure keys on Freenet. The *Name* can be assigned freely with one or two restrictions, and it is not related in any way to the file content. The disadvantage is that multiple users can assign the same name to different file content, thus causing collisions and inconsistencies.

Updateable Subspace Keys (USKs)

USKs are similar to SSKs. Their elements are identical, except for the *Version number*, which serves the same purpose as in the SSK, except that the Freenet node will automatically find the latest version of the USK.

In typical newsreader style, Frost divides the main window into three panels (see Figure 5). Links in forums you subscribe to are displayed in a directory tree: Messages from the selected forum are shown top right; below this you can see the text for the selected message.Some messages are fairly long because Frost users tend to use full-page quotes in their responses. Because the source they are quoting may not be accessible on Freenet, the quote often contains the whole message to preserve the original context.

The forum area is located in a tab, with separate tabs for uploads and downloads and a simple file sharing mechanism. Above these tabs are buttons for configuring Frost and organizing the forums you subscribe to. The most import button is the one with the globe icon. It opens an overview that lets you select and subscribe to known forums.

Publishing Pseudonyms

A column labeled *Sig* appears along with the forum. For messages posted by users with pseudonyms, the *Sig* column contains a note on the user's trustworthiness. It is up to you who you trust. The buttons above the list of messages let you specify the degree of trust in a pseudonym. This feature is useful because you can configure the reader to show you messages as of a certain level of trustworthiness. The pseudonyms start with an unknown level of trust (*CHECK*). *BAD* is the right setting for trolls, and *GOOD* for more pleasant Freenet inhabitants. *OBSERVE* is somewhere between *CHECK* and *GOOD*.

Messages can contain references to other forums, Freenet keys, or file attachments. Frost lists attachments at the end of the message, and it displays Freenet keys in the text as hyperlinks. Users can right-click a link to add it to the *Download* tab.

Basis for Extensions

Freesites and Frost are just two examples of applications based on Freenet. Some

INFO

[1] Freenet homepage: *http://www.freenetproject.org/*

[2] Frost homepage: *http://jtcfrost.sourceforge.net/*

[3] Arch on Freenet: *http://www.unix-ag.uni-kl.de/~conrad/Archives/DSDiF/*

other Freenet applications include the Freemail mail tool and the Thaw file sharing utility.

Work is in progress on other applications, such as an NNTP gateway or adaptations of version control systems such as Mercurial or Arch [3]. A streaming mechanism is on the map for a future version of Freenet.

The Freenet node also includes an interface for plug-ins and a network interface for client programs. Developers are free to extend the model and integrate it with existing programs.

Price of Freedom

Freenet is licensed under the GPL. The current 0.7 version of Freenet works extremely well, despite its complexity; it is easy to install and well documented. The controls are simple, and the system offers a high degree of security.

Freenet has made much progress since version 0.5, which is still very much in active use; however, this progress has had an effect on upload and download performance for volumes of data above 100MB.

Despite excellent usability, Freenet is still a very complex system that requires serious attention to the topic of anonymity on the part of the user.

If you compare a Freesite's performance with that of an ordinary website, you are bound to be disappointed. But performance is not the most important consideration of the Freenet community. It is Freenet's declared goal to ensure a free exchange of opinions and information even in unfree environments.

Conclusion

No one can deny the need for this kind of tool, and on the basis of Freenet's popularity, it looks like the Freenet network has the potential to serve the role of building the foundation for an anonymous Internet. ■

Security

Freenet guarantees the confidentiality, anonymity, and authenticity of the data it manages on various levels:

- Freenet encrypts files on uploading and does not decrypt them until downloaded – the files stored on a node are thus invisible to the operator. This allows the operator to plausibly deny any knowledge of the files.
- Nodes use encrypted communications – no external third party can discover who requests a key or stores its data.
- Freenet pads files to fixed size packages – the file size and the number of data packages exchanged do not allow any conclusions on the file content or the path the data takes through the network.
- Each node only sees its immediate neighbors – no node can tell whether an incoming request originated with the neighbor node that was the immediate source or whether the request is simply being forwarded.
- When a user requests a file, Freenet copies the file multiple times en route through the network – if a participant switches off his node, this does not necessarily mean that the data stored on it will be lost.
- Nodes can be configured to connect only to specific nodes, such as nodes run by friends and acquaintances – this means that only trusted persons learn that a participant actually runs a node. A network comprising trust-based connections is referred to as a darknet by the Freenet community.