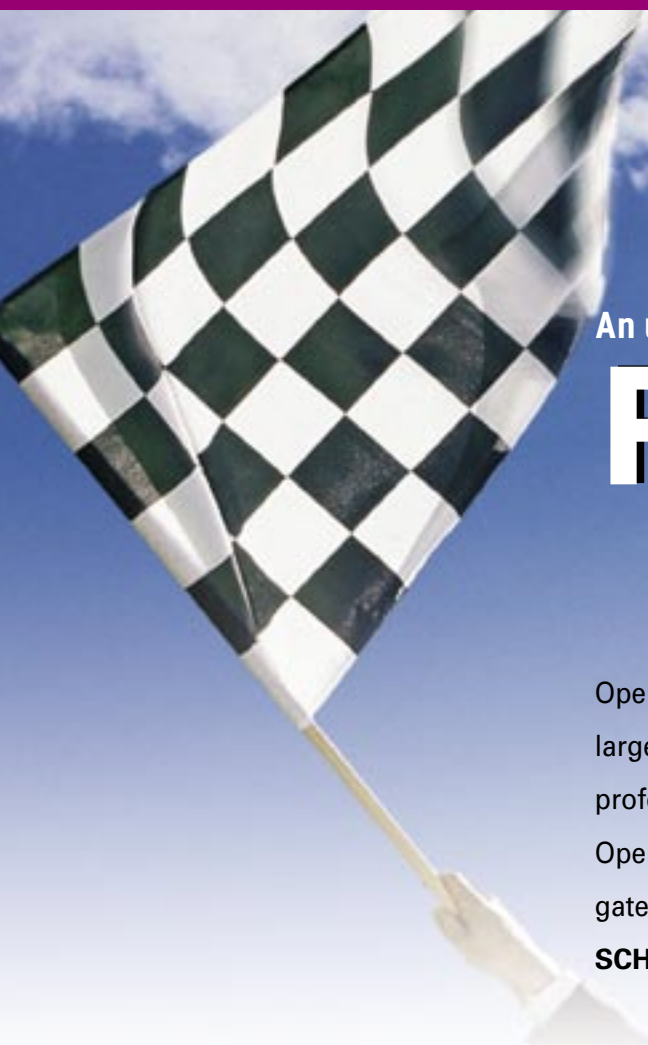

An up-to-date look at free software and its makers

PROJECTS ON THE MOVE

OpenWRT puts Linux on WLAN routers and helps users set up large-scale WLANs at home, and the FreeWRT derivative adds a professional touch. If you don't have your own compile farm, the OpenSUSE build service may be just what you need. And we investigate the obstacles to new packages for Debian. **BY CARSTEN SCHNOBER, ANDREA MÜLLER, AND MARTIN LOSCHWITZ**

Wireless LANs have helped many home users to set up simple networks without having to run cables through the house. Typically, a router handles shared Internet access for the internal machines. Many DSL providers give users discounts on WLAN routers, or even give away a router. Low power consumption, a small footprint, and low noise emission make routers infinitely preferable to configuring a PC for the same job.

Routers between PCs and Embedded Systems

One problem with routers is the lack of functionality. The default firmware normally doesn't support anything much but the core task of routing between the Internet and the internal network. Although many routers now support peripheral tasks such as port forwarding, you can't expect much more than that. And if you intend to run any other kind of software on your router, you will need an operating system to match.

Thankfully, there is no need to replace a practical router with a PC, as many of these small-footprint machines will actually run Linux. The OpenWRT [1] distribution has helped to put Linux on a growing number of WLAN and Ethernet routers since 2004.

The OpenWRT project was originally based on work done by Linksys who supplied their WRT54G router (Figure 2) with Linux as the factory default firmware. The first version of OpenWRT was based on this, although support has now been extended to devices other than Linksys; the OpenWRT homepage [1] has a full list.

To compensate for the lack of storage capacity on the router's minimal system, OpenWRT uses the lean uClibc [2] library and Busybox [3], which implements a shell and a number of command line tools in a resource saving manner. Ready-to-run OpenWRT packages include SSH, Telnet, and an SSH server. Lighttp [4] provides an optional web-based router configuration interface, and of course it will serve up other web pages. If you need to prioritize time-critical applications, the good news is that OpenWRT supports traffic shaping, an ability that legacy router firmware will not normally give you.

Free Communities

Various other projects have used the good work put in by OpenWRT to design routers for special tasks. Mesh networks are a popular application: in a mesh network, each access point acts as a repeater in a decentralized network forwarding requests from other attached nodes to the internal network, and to the Internet. The Freifunk [5] initiative adds the OLSR protocol to the standard OpenWRT installation to support mesh networks.

The Freifunk project is a non-commercial initiative that aims to provide free public access to the Internet as a communication medium via free networks. To help achieve this aim, the idea is to provide an alternative to commercial Internet providers by setting up a community-owned network infrastructure in your local district.

Of course, these community-owned network structures will still rely on Internet providers to give the internal network access to the Internet, but just a few private broadband connections as interface points are all you need, if the mesh network can provide a reliable connection.

Foneros!

Fon [6] takes a similar approach, but with a commercial background. Just like

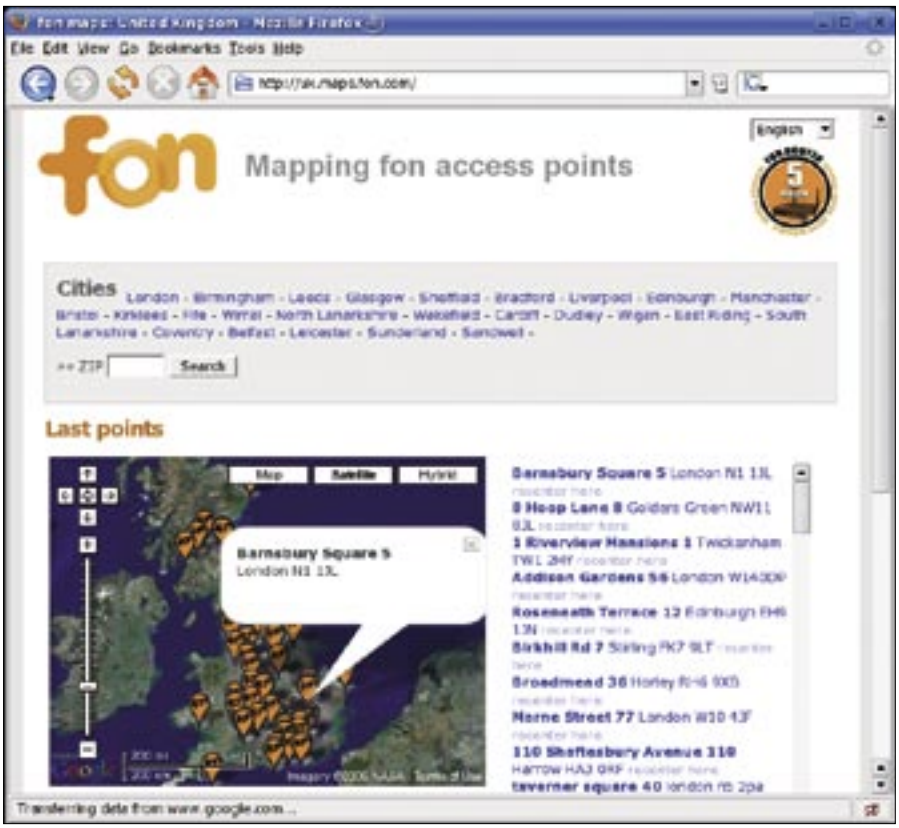

Figure 1: Fon is trying to set up a global WLAN. Wealthy sponsors provide funding to ensure continual growth.

the Freifunk initiative, the Fon project uses routers with OpenWRT, to set up a global system of WLAN hotspots.

At present, Fon is still in the beta phase, and is looking to set up as many access points as possible. After initially focusing on the USA and Spain, Fon now has a number of access points in the UK (see Figure 1). Fon offers new users a version of the Linksys WLAN router referred to previously, pre-configured for use as a Fon hotspot, for $US 5 in the US, or 5 Euros in Europe.

Getting Organized

There will be three categories of so-called Foneros in future: some Foneros give other Fon users free WLAN access to their Internet connection. An Alien is a Fonero without an Internet connection of their own who pays to use Fon hotspots – prices will be somewhere in the region of two to three Euros for 24 hours. The third Fonero profile is known as Bill, and refers to users who charge Aliens for using their Internet connections. Bills pay for access to other Fon hotspots. As of this writing, the Fon network does not have a billing system in place: Bills and Aliens are future, and the only existing nodes are run by Linus Foneros.

The fact that today's Fon hotspots have a short range restricts their use to Foneros who live in the neighborhood or happen to be passing through; and this is far-removed from full cover. This said, with financial backing from major players such as Google and Skype, it certainly looks as if Fon could achieve its high-flying goals in the long term, with sponsors subsidizing purchasing of new routers.

The legal situation for Foneros is uncertain. In contrast to people who rely on the Freifunk initiative, the commercial side of the Fon network may mean that they are subject to the same regulations as Internet providers, and this is something that definitely needs to be clarified before the billing system is deployed.

One obligation that commercial Internet providers in various countries have is to allow the authorities to wiretap communications. Additionally, the EU is looking to introduce legislation to force providers to log and store all their traffic for a period of several months to support law enforcement. As this would be more or less impossible for a home user, Foneros would be forced to give up, paving the way for non-commercial initiatives such as Freifunk to set up large-scale volunteer community WLANs.

Forked

Although OpenWRT is fairly widespread, the OpenWRT project now faces the same fate as many other open source projects. Friction between the developers has led to the project being forked.

Long-serving OpenWRT developer Waldemar Brotkorb posted an open letter to the OpenWRT community [7] stating poor internal communication and public presentation as his reasons for launching the FreeWRT [8] project based on OpenWRT.

According to Brotkorb, FreeWRT will be aiming to set up a more interesting and professional project homepage, and it will attempt to achieve more reliability with regular releases and less dramatic changes between versions. In future, users will be able to compile new firmware images for their routers on Linux, Windows, and Mac OS X; the FreeWRT update mechanism will aim to keep existing settings.

The FreeWRT founder will be looking to make life easier for developers and new software packages by introducing understandable policies. According to Brotkorb, the focus of FreeWRT development will not be on the goal of implementing new features and supporting

Figure 2: Linksys was the first vendor to supply a router with Linux firmware, forming the basis for the OpenWRT distribution.

new hardware, but on achieving stability and reliability.

Brotkorb will not be quitting OpenWRT development completely, and can't promise that FreeWRT will achieve all of its high-flying goals. Instead, FreeWRT will be an alternative to OpenWRT, especially for use in professional environments. He also stated that companies such as Linksys and Cisco were interested in the project. The ability to backport bugfixes will mean that both of these parallel projects will benefit from one another's work.

Do-It-Yourself

Free software developers usually aim to help those interested in their work by providing ready-to-run packages. The ability to install an RPM, without having to resolve dependencies manually, or worry about uninstalling, might be just what it takes to convince a hesitant user to install.

But private developers rarely have access to a compile farm with the various architectures and distributions needed to produce clean RPMs. The OpenSUSE project is looking to make life easier for free developers by providing the OpenSUSE build service at [9]. Registered users can upload source code packages and a special RPM spec file, and the server takes care of creating RPM packages for Suse, Fedora, and Mandriva. Developers can manage their build projects via a web interface, use the OSC command line tool, which is available as a download on the build service homepage, or use tools of their own that leverage the build service API.

The spec file for the build service is similar to the spec file for the normal RPM build process, but with a few build service-specific keywords added. The build service gives developers RPMs compiled on 32 and 64 bit PCs with defined dependencies for various distributions. Users benefit from the fact that the build service gives them access to up-to-date packages.

As of this writing the build service is still in the alpha phase, but despite this it can already handle the core tasks. Developers can register by mail. OpenSUSE is looking to go beta in the third quarter of 2006 when the full feature set will be implemented and ready for bug squashing.

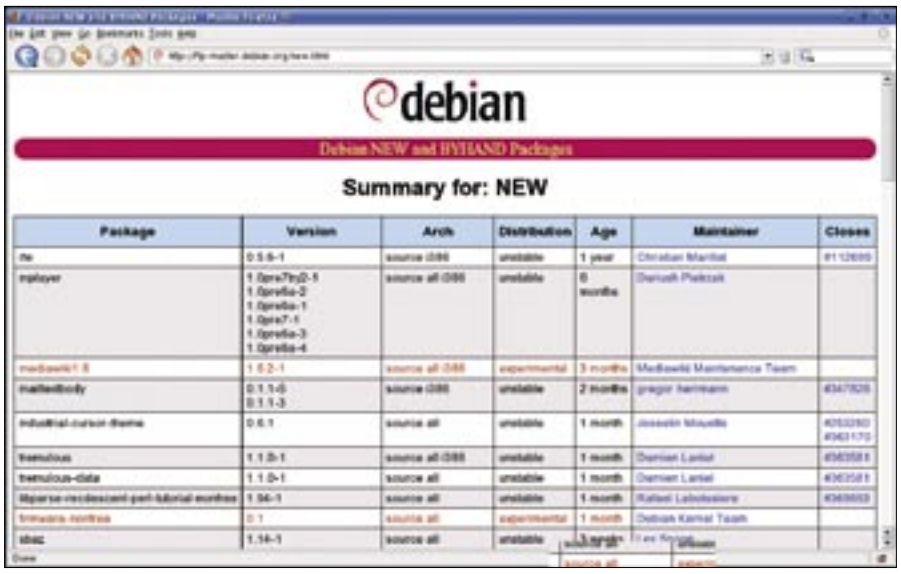

Figure 3: Waiting in line in the New queue – part of the obstacle course newly submitted packages have to negotiate.

If the build service works as intended, OpenSUSE will be looking to add more features, for example, support for package formats other than RPM.

Obstacles to New Debian Packages

New packages for Debian follow a well-trodden path. After uploading to the archive, they first enter the *sid* developer branch, are passed to the *testing* branch, and finally land in *stable*. When a developer uploads a package that was not previously available in its current makeup, or if the package is completely new, it first has to go through the *New* queue.

This is necessary even from a legal point of view. You may remember the so-called *Non US* archive which held software with cryptographic functions up to a few years ago. The reason for this was the export embargo on encryption software in the US at that time.

The Labors of Sisyphus

Officially, the FTP Masters are responsible for the *New* queue where they check incoming packages and wave them through if all is well. There is no way of automating this process; and this is the crux of the problem. As many FTP Masters do not have enough time, there is often a backlog of 250 or more packages in the *New* queue (Figure 5). This annoys the developers because their work has often lain dormant for weeks or even months before it even reaches a distribution.

On the other hand, the FTP Masters will pass key packages if requested to do so by the Release team. This said, most packages are forced to wait in line while the package flow for the unstable branch, and thence to testing, runs dry. Periods of weeks where nothing happens are quite common.

Just recently, Debian developer Jörg Jaspert, who joined the FTP Master project last year to offload some of the work, cleared some of the backlog. But there is no way that Jaspert can shoulder all of the responsibility for the *New* queue on a long-term basis.

Many Debian developers criticize the fact that although the FTP Master team has many members on paper, some of them seem to have gone into hiding. The *New* problem is starting to slow down the whole operation. ■

INFO

[1] OpenWRT: *http://openwrt.org*
[2] uClibc: *http://uclibc.org*
[3] Busybox: *http://www.busybox.net*
[4] Lighttp: *http://www.lighttpd.net*
[5] Freifunk (German): *http://freifunk.net*
[6] Fon: *http://www.fon.com*
[7] Open letter to the OpenWRT community: *http://openbsd-geek.de/archives/224-An-open-letter-to-the-OpenWrt-community.html*
[8] FreeWRT: *http://www.freewrt.org*
[9] OpenSUSE build service: *http://build.opensuse.org*