BSD in the routing industry

Massimiliano Stucchi

stucchi@gufi.org
Italian FreeBSD Users Group
http://www.gufi.org

Abstract

The BSD family has always been very well known for its robust network stack, hence it has been widely used in many different fields and applications. In the ISP market, though, the situation is totally different, and solutions employing *BSD operating systems are often discarded in favour of proprietary solutions.

In this talk we will discuss the different possibilities offered by the BSD operating systems family in terms of networking tools and practices, compared to proprietary solutions offered by companies such as Cisco and Juniper, detailing the differences between them and highlighting the major points and drawbacks of each of them, up to a cost comparison in real field applications.

Real field applications will be introduced via explanation of the solutions created using BSD-based routing software in the real industry running in two different environments, an ISP spanning Europe and another one offering WISP services.

We will also delve into the experience in running a FreeBSD-/OpenBSD- and OpenBGPd-based route server at MINAP, the MIlan Neutral Access Point (*http://www.minap.it*), describing success stories and guiding the audience into a comparison with the other route servers running at the same IX, powered by Linux and Bird/Quagga.

Introduction

Even though we all now of the superior-ness of the BSD implementation of TCP/IP over any other, we are also aware that this fact is not really appreciated in the Carrier/ISP market, where, most of the time, proprietary solutions are preferred. This is due to many factors - commercial support as the most predominant one - that make the decision “simple” for management, and give a hard time to OSS advocates. Economical aspects are often secondary, and are often superseeded by fact sheets demonstrating high MTBF (Mean Time Between Failures) from hardware vendors, as well as support SLAs - which do not match reality often.

Being an advocate myself, and having been in discussion with customers about these many solutions, helping them decide which one to adopt, and following them in their dailiy activities gave me a wide understanding of the whole area. I’ve also had to make presentations pushing the customer towards the most viable, simple, elegant and economical solution, and this (very) often meant advocating the BSDs and their software, and being able to support with data what I was pushing towards.

The BSDs can be a very efficient solution, bringing lots of scalability and chances to customize the product. Something we don’t easily find in proprietary solutions. Needless to say, there are many different facilities and software to create a carrier-class network using any *BSD operating system.

BSD Routing facilities

There are different facilities and softwares on the BSDs, either as part of the base system or as a port.

Any of these can be effectively used in production environments, and have its own advantages and disadvantages. Here’s a list of them, with a brief description:

OpenBGPd: As the name says, it was created as part of the OpenBSD operating system. It's the most flexible daemon, supports atomic configuration reload, and can be used in heavy load environments, as well as in route servers, but we'll see this applied later in the paper.

OpenOSPFd: Similar to the OpenBGPd daemon, it differs in the protocol supported. It works well in conjunction with the BGP daemon, and it's really effective to be used as an IGP daemon protocol.

Quagga: Born from the Zebra project, it's the oldest in this family of daemons. However, its architecture reflects its age, and there have been rumors of a planned rewrite to fix some of the problems which are making users move away from using it.

BIRD: Is the last on the list and last in the implementation. Work on the software is financed by the Czech NIC labs. Its flexibility and capacity of running under heavy load and with a large number of peers are encouraging many IXPs to move to it from Quagga.

All of these software support both IPv4 and IPv6, as well as AS32.

Different perspectives

As we could see from the list of software, there are many choices on the BSDs, if you want to use them for your routing purposes. However, pfSense is the project we can most often hear about. This is not a problem per-se, but pfSense is aimed at those small to medium businesses and SOHOs where a small appliance/router is needed, and where performance is not a problem. This is not the case for bigger businesses, carriers and ISPs, where performance is an issue, and where the BSDs are often discarded, giving precedence to well-known vendors and their solutions.

So, the point here stands in presenting the different solutions, taking all of them into account and not discarding any. There are good reasons to go opensource, and they go beyond economical reasons.

A reason to go OpenSource

As mentioned earlier, there are more reasons to go opensource, other than economical ones. We'll start from the economic aspect, though.

Going for the perfect solution is a matter of combining the right hardware and software. Choosing one or the other is not a matter of knowledge, as the protocols used are the same, but it's a matter of knowing about the alternatives, and - sometimes - it may also be a matter of finding the right interface cards. ATM is, for instance, one of the cases where going with a custom-built solution using the BSDs is not feasible, since there are not ATM cards on the market equivalent to the ones you are able to find from any of the hardware vendors such as Cisco or Juniper.

For instance, though, a solution for a router able to handle two to three full routing tables, with capacity to handle around 200mbits of sustained traffic, can cost one tenth of the same equivalent from either one of the hardware vendors out there, whereas it drops down to "only" 1/4th using refurbished hardware.

We also have to keep an eye on the features. The BSDs have everything you might need, out-of-the-box, or just one installation away, while adding a functionality to a closed-box might need paying an extra for a different feature-set and having to reboot the device to upgrade. There is also the case, for instance, of the AS32 support. It is also present in any software, but not on the Cisco platform, at the time of writing. This means that in case you should have to peer with a company with a 32-bit AS number, you will have to refuse, not being able to service it.

The opposite goes for MPLS support, which is not present on the BSDs at the time of writing. Work is under way to write it, soon.

The MINAP Project

MINAP is a project run by a few ISPs and carriers in the Milano-Caldera campus. The goal is to create a distributed IXP to be parallel to the single-facility-based MIX (Milan

Internet Exchange - *http://www.mix-it.net*), which has suffered of problems in the past, affecting the whole Italian routing and thus creating a single point of failure in the whole Italian internet industry. MINAP is a free access IXP, and relies on donations of hardware and time from its members. Differently from the single PoP MIX has, MINAP evolved into a distributed set of PoPs in the Milan-Caldera Campus, a place where nearly 90% of all the international routes transiting Italy converge, making it easy to join for any national and international company wanting to.

Route servers at MINAP

One of the initial goals was to make MINAP the first Italian IXP to have a route server. A route server is a device running a BGP daemon establishing a session with each peer, and forwarding any route information received to the whole IXP network. This type of device helps in reducing the amount of work needed on a member's configuration any time a new member joins or leaves the IXP and when there is any type of modification in the routes announced over the IXP. This way there is no need to have particular filters in place on a member's session, whereas there is some more work for the exchange's staff.

Experiences at other IXPs

With this in mind, we started looking at solutions and implementations at other IXPs, trying to get a clear idea of any advantages and disadvantages for one solution over the other. We looked mainly at the biggest european IXPs such as LINX and AMS-IX, where we could see that different software was used and implemented on different operating systems.

Out of our research, we noted that the two aforementioned IXPs were both running Quagga at the time, but both experienced repeated problems and both had an on-going project aiming at moving away from it. AMS-IX completed the migration first, and moved to OpenBGPd around mid-october 2009. LINX, instead, decided to migrate to BIRD, and the move is on-going at the time of writing. Problems with Quagga are related to repeated crashes in environments with more than 200 peers. Moreover, sessions could not be restored automatically after the daemon had been restarted, as this would have caused the daemon to crash again repeatedly. Experience at LINX showed about a crash weekly accross the four route-servers, with a few collective crashes. Any time this happened, manual intervention was required in order to re-establish peering sessions one at a time, keeping staff busy for about two hours every time this happened.

MINAP does not have so many peers, but we wanted to choose a platform based on long-term stability predictions, and Quagga didn't really seem a viable solution. In the end, we decided to implement a route server based on FreeBSD/OpenBGPd and one running Linux/BIRD.

This has been a solid solution so far, meaning we can reliably grow over time as a project without having to worry and invest time in fixing problems.

Conclusions

We have highlighted how, using BSD-based routing solution, a company can not only reduce costs, but also have greater stability and better features supported accross its network. The example of MINAP and the other internet exchanges has given an idea on how important players in the internet world are benefiting from using this kind of solutions over others to increase reliability and functionality for everybody using the internet today.