

An interview with Hideaki Yoshifuji

Mauro Tortonesi

Deep Space 6

`mauro@deepspace6.net`

Peter Bieringer

Deep Space 6

`peter@deepspace6.net`

Hideaki Yoshifuji (mailto:yoshifuji_at_linux-ipv6.org) is the most active developer working in the USAGI project (<http://www.linux-ipv6.org>), an important development initiative aimed at improving the IPv6 support of Linux, both in the kernel and in userspace.

He is a student pursuing his Ph.D in Information Science and Technology from the University of Tokyo (<http://www.hongo.wide.ad.jp>). His major research interests are IPv6, network management, security, CDN, and operation systems.

In this very interesting interview, Hideaki discusses with us the significant goals already achieved by the USAGI project, the latest improvements they've made to enhance the IPv6 support of both the Linux kernel and userspace apps, and presents an ambitious future development roadmap.

Hi, Hideaki. First of all, thank you very much for having granted us this interview.

You're welcome. It's also my pleasure to have this interview.

As everyone of our readers know, the USAGI project is a japanese initiative for improving the IPv6 support of Linux (including the kernel, the glibc libraries and the userspace applications). The USAGI project is sponsored by several japanese corporations (among which there are even big names like Hitachi and Toshiba) and research institutes like the University of Tokyo - for which you work. How did you get involved in this project?

I am one of the members who established our project. (NOTE: Hideaki is also a core member of the USAGI project).

In the last two years, the USAGI project has produced a lot of valuable improvements to the Linux kernel. You have added many features (like source and destination address selection (<http://www.ietf.org/rfc/rfc3484.txt>), privacy extensions (<http://www.ietf.org/rfc/rfc3041.txt>), ISATAP (<http://www.ietf.org/internet-drafts/draft-ietf-ngtrans-isatap-12.txt>) support, anycast (<http://www.ietf.org/internet-drafts/draft-ietf-ipngwg-ipv6-anycast-analysis-01.txt>) support, Node Information Queries (<http://www.ietf.org/internet-drafts/draft-ietf-ipngwg-icmp-name-lookups-09.txt>), double binding (<http://www.ietf.org/internet-drafts/draft-ietf-ipngwg-rfc2553bis-10.txt>), dropping of fake ipv4-mapped addresses (<ftp://ftp.itojun.org/pub/paper/draft-itojun-v6ops-v4mapped-harmful-01.txt>), etc...) and enhanced or debugged existing ones (like neighbour discovery (<http://www.ietf.org/rfc/rfc2641.txt>), v6 netfilter (<http://netfilter.samba.org>), tunnels (<http://www.ietf.org/rfc/rfc2893.txt>), etc...). You have also integrated in your kernel external code to provide IPSEC (<http://www.ietf.org/rfc/rfc2401.txt>) and Mobile IPv6 (<http://www.ietf.org/internet-drafts/draft-ietf-mobileip-ipv6-21.txt>) support. It is a pretty impressive work. What other features are you planning to implement? Should the users expect to see major improvements in USAGI's code along 2003?

We are planning to complete our implementation of the IPsec and Mobile IP protocols, and merge it into the mainline kernel.

When is the next stable release scheduled for?

The next stable release, USAGI STABLE 4.1, is scheduled on the St. Valentine's day :-) (NOTE: this part of the interview was made before Feb 14th 2003) Then, we'll make a new stable release in summer.

The USAGI project developers have also spent a lot of effort to port the main Linux userspace networking utilities to IPv6 or to enhance their IPv6 support. What is the future USAGI development roadmap in the userspace applications' field?

Streaming applications including VoIP.

Recently, you have written two new userspace applications: ninfod and ziconfig. Can you briefly explain us what these applications are about?

ninfod is the Node Information Daemon, which implements a responder for Node Information Protocol queries. The Node Information Protocol is described in draft-ietf-ipngwg-icmp-name-lookups-09.txt (<http://www.ietf.org/internet-drafts/draft-ietf-ipngwg-icmp-name-lookups-09.txt>). We had first implemented that function in kernel space, but we decided to re-implement it in userspace for the following reasons:

- we needed a more flexible configuration mechanism, including ACLs (but this is not implemented, yet)

- kernel implementation was really too big.

ziconfig instead is a trivial configuration tool for scoped routing support. (But we don't have zone-id in `sin6_scope_id` in `sockaddr_in6` support, yet.)

What part of the code that USAGI has produced are you particularly satisfied with? Is there something that in your opinion still needs to be worked on (apart from MIPv6 and IPSEC)?

I'm happy with IPsec and Node Information Queries. We still need to work on Multicast routing / MLDv2 (<http://www.ietf.org/rfc/rfc2710.txt>).

Have you ever worked on a *BSD kernel? Do you think that the IPv6 support of FreeBSD, NetBSD and OpenBSD (which is developed by the KAME project (<http://www.kame.net>)) is overall better than the Linux one?

Before we started, there were many bugs and missing features in the Linux IPv6 support and, therefore, *BSD*s' IPv6 support was much better than Linux's one. Well, now severe bugs have been fixed, and missing features such as IPsec and/or Mobile IP (this is HUT (<http://www.mipl.mediapoli.com/>)'s work) have been implemented. So, it probably depends on your taste.

***BSD have rather full KAME-IPv6 included in mainstream in contrast to Linux, where the most important enhancements are only available via USAGI's kernel patch. What's your opinion whether this delays full IPv6 usage with Linux?**

Since we contributed critical fixes for normal IPv6 usage with Linux, we believe that people can use IPv6 with Linux as BSD-people do.

As the most representative USAGI developer, you have also worked with the Linux kernel networking-stack development team. You have submitted several patches to fix many broken behaviours of the vanilla kernel (e.g. double binding), but these patches have been accepted only for the 2.5 development series (where they have already been merged). Why haven't these patches been merged in the stable 2.4 series?

It seems that some of them will appear in 2.4.21. (We can see them in 2.4.21preX patches.)

During the last months, many Linux users expected to see a major merge of the USAGI code in the kernel mainstream, but this has not happened. What's the reason, in your opinion?

We have finished working on most of the major features/improvements, though IPsec, source address selection, and privacy extensions support are still pending. The reason why other small fixes were not

going to mainstream kernel is because we focus on making new IPsec implementation for linux-2.5. Another reason is I was busy...

What is the status of IPv6 support in the Linux 2.5 kernel series? The feature freeze for Linux 2.6 has been already reached, but it seems that there hasn't been the significant improvement of the IPv6 stack that was scheduled.

Our goal of Linux 2.6 kernel series is supporting IPv6 IPsec. In the future releases later than 2.6 series, we'd like to support multicast routing, mobile IP and netfiltering.

We'll continue working making small patches for mainstream kernel in February. I suppose 2.5 (or 2.6) will be as almost the same quality as ours (except for IPsec and/or Mobile IP - this is HUT's work though).

Have you talked with the other kernel developers about this? Is there a credible development roadmap for IPv6 support in Linux 2.6/2.7? Perhaps a plan for the merge of USAGI code?

Yes, we attended to Ottawa Linux Symposium 2002 (<http://www.linuxsymposium.org/2002/>) and received many helpful comments. (NOTE: Hideaki is scheduled to talk at the next Ottawa Linux Symposium 2003 (http://www.linuxsymposium.org/2003/view_abstract.php?talk=62)).

Do you know any Linux distributor which plan to integrate USAGI patches in their patched kernels some day? Or will they wait until vanilla kernel contains USAGI extensions?

There're several distributors which have already shipped with USAGI. For example:

- Momonga GNU/Linux (<http://www.momonga-linux.org>) (a.k.a. Kondara GNU/Linux)
- SSD/Linux (<http://openlab.plathome.co.jp/linux/ssdlinux.html>)
- Debian GNU/Linux (<http://www.debian.org>) (patch package is available)
- Omoikane GNU/Linux (<http://www.omoikane.co.jp>)
- Miracle Linux (<http://www.miraclelinux.com>).

Our efforts are being used more widely as our extensions are being integrated to mainline kernel, you know.

Have you followed the development of glibc? Will there be an improvement of IPv6 support for RPC and the Extended BSD Socket API in the next releases?

Not so much because of lack of human resource. It is one of our TODOs.

What is the internal organization of the USAGI development team? Is there a Linus-like benevolent dictator in the team who takes the final decisions or do you work by consensus?

By consensus. We hold face-to-face meeting once a week.

You are surely the most active of the USAGI project's core developer. Frankly speaking, it is from Alan Cox's glorious days (when the whole World believed that there were at least three developers behind that name) that i haven't seen a developer working so hard on the Linux kernel ;-) What is the secret behind such an impressive productivity?

I am not a person, but a team like Alexey :-). (NOTE: Hideaki here is talking about Alexey Kuznetsov, who is the author of *iproute2* and surely one of the best developers of the Linux networking stack.)

What suggestions can you give to a newbie developer who wants to begin working on the Linux networking stack? Where should he start from?

Hackers never sleep :-). Actually, read codes, put *printf*'s everywhere.

Some Unix server programs still lacks of full or stable IPv6 support, e.g. Apache2 (<http://httpd.apache.org/>) or Squid (<http://www.squid-cache.org/>). What are the reasons, in your opinion? Are there too few IPv6 developers and testers available to help the projects?

As far as I know, Apache 2 works well. I don't know about status of IPv6 support in squid because I haven't use squid for a long time. This would be a problem; there're few IPv6 developers / testers available to help them.