Implementing IPv6: experiences at KAME project

Jun-ichiro itojun Hagino <itojun@iijlab.net>

Research Laboratory, Internet Initiative Japan Inc.
http://www.kame.net/

Abstract

KAME project was started in 1998 to provide BSD-licensed reference implementation for IPv6, IPsec and other recent Internet technologies. This paper tries to describe our motivation, where are we right now, technical hurdles we have faced and solutions against them, and our future directions.

All of our source code has been, and will be distributed under a BSD-like license.

1. Motivation, and the history of IPv6 and KAME

At IETF (Internet Engineering Task Force), IPng effort was started in 1992 to re-architect IPv4, and make it possible for the Internet to grow much further [Bradner, 1993] . IPv4 was designed in 1970s, and IPv4 address has only 32bits. 32bits may be sufficient for designs in 1970s, but not for 1990s and beyond. The theoretical limit of IPv4 address space is 4.2 billion, which is much less than human population on the planet. Actual upper limit is much smaller, due to the fragmented allocation of IPv4 address space - tens or hundreds of millions. With the current trend of "everything over IP", cellphones, microwave ovens, every single motorcycles will get connected by IP. For 1990s and beyond, IPv4 design is clearly insufficient.

In 1995, the final candidate for IPng was selected, which is now called IPv6 [Deering, 1995; Deering, 1998] . We, WIDE project (a research consortium for Internet technology, consists of 100s of universities and companies), thought IPv6 is the way to go, and implementation efforts were started. By 1996, within WIDE project we had 7 independent implementations of IPv6.

The independent implementations were good as a start (perform interoperability tests, identify common implementation errors, and identify protocol flaws), however, as time goes by we faced two problems. The first one is a human resource problem - each of the implementation had only one or two dedicated coders, and they do not have enough time to maintain their codebases. The second one is deployment problem - we felt an urgent need to integrate IPv6 stack into various BSD distributions, and to redistribute it under BSD license.

In IPv4 days, BSD IPv4 stack was considered as a reference implementation. Interoperability tests were made against BSD IPv4 stack, people studied IPv4 from the code (like Stevens' "TCP/IP illustrated" books), BSD IPv4 stacks were deployed to confirm the scalability/dependability, and it improved the IETF standards. On the contrary, IPv6 was started with no (or very few) testbed codebase, there was no reference implementation, and we had almost no experience in deploying IPv6 codebases to wide-area network (not a tiny laboratory network).

Therefore, in fall 1997, we decided to start an effort to (1) integrate the codebases we have into one, to make it a single effort, (2) redistribute it under BSD license, (3) operate the code nationwide to confirm the scalability/dependability, and (4) integrate our codebase into *BSD operating systems to redistribute it widely and to make it a reference implementation. This is what the KAME project is about. KAME project itself is focused into the implementation (coding) part of the above goals. Network operation part is handled by IPv6 working group in WIDE project. We also have other sub-projects in WIDE for IPv6 conformance testing (TAHI project) as well as IPv6-for-linux project (USAGI project).

2. Status of the project

KAME IPv6/IPsec stack is integrated into all 4 *BSD projects (NetBSD, OpenBSD, FreeBSD and BSD/OS), as well as various vendor router products (like Juniper, Extreme, Hitachi and Fujitsu), commercial operating systems (Apple MacOS X) and embedded products (like Windriver's VxWorks). I think it is safe to say that we are successful in making a reference implementation of IPv6.

2.1. The integration policy

Though IPv6/IPsec basic specs are set in stone, there are hundreds of internet drafts coming up for IPv6 and IPsec. There is a questions on how we should integrate these portions into *BSD.

We basically have two separate tracks - *BSD track, and KAME track. *BSD track has protocol supports for specs made it to RFC, and are needed for wide availability. KAME track implements various internet drafts so that we can identify protocol flaws. Some of the protocol drafts are really experimental, and needs to be torture-tested.

For instance, Mobile IPv6 is yet to become an RFC, and every time a new revision of internet draft is issued, header format and other things are changed. Therefore, it is not integrated into *BSD releases. Mobile IPv6 code is available in KAME distribution, which is distributed as a patch to the latest *BSD releases.

3. Technical hurdles in IPv6 implementation

To make IPv6 successful, IPv6 was designed to follow the same design principles of IPv4. So basic portion of IPv6 implementation can be done without any trouble; we just need to follow IPv4 practices. Still, IPv6 includes many functionality enhancements and design corrections from IPv4, to take advantage of experiences gathered between 1970s (when IPv4 was designed) and 1990s. Some of the enhancements caused technical difficulties on us implementing IPv6 into BSD network protocol stack. In this section we would like to discuss some of them.

3.1. Multiple addresses on an interface

In IPv4, it was assumed that single IPv4 address is associated with an interface. Actually an IPv4 address identifies a node, not an interface. With IPv6, it was clarified that an IPv6 address is associated with an interface (not with a node), and it is assumed that multiple IPv6 address would be assigned to a single interface.

In the design of BSD IPv4 stack, there are assumptions made to take advantage of the above. For instance, routing table entries hold a cache of source address used for the particular peer (rt_ifa member in struct rtentry). The use of this field is designed based on IPv4 assumption. For IPv6 code, we do not use the cache for source address selection. We invoke source address selection logic documented in [Draves, 2003] instead.

3.2. Kernel stack concerns with extension header chains

IPv6 replaced IPv4 option headers by extension header chains. With IPv6, multiple extension headers can be attached to a packet, constructing a structure that looks like a linked-list. There is no upper limit defined for the number of extension headers, protocol-wise. If we apply the traditional BSD protocol dispatch logic, we would make a function call every time we see an extension header. It will result in a deep function call chain, and possibly a kernel stack overflow. To address this problem, we have implemented a dispatch logic which does not create a deep function call chain [Hagino, 2000] .

3.3. Securing IPv6 protocols and APIs

While we are working on IPv6 implementation, we have found vulnerabilities, or potential vulnerabilities, in IPv6 protocol and API specifications. To provide a brief summary, we have identified problems like below:

- Some of the IPv6 transition technologies can be abused by malicious parties for transmitting network traffic anonymously, or for bypassing access controls [Hagino, 2000] .
- Some of the IPv6 transition technologies introduce ambiguous interpretation to the peer's identification to the applications; an IPv6 address provided to an application can have two different interpretation [Hagino, 2003; Metz, 2003] . Malicious parties can confuse an application and bypass access controls by taking advantage of the confusion.

We have also provided large number of feedbacks to IETF standards from our implementation experiences.

4. Towards the real IPv6 deployment

There are a couple of roadblocks against IPv6 deployment at this point:

- More ISPs needs to become IPv6 ready, and start providing services, otherwise the customers cannot use it. ISPs needs to act proactively in deploying IPv6 network (rather than waiting for customer demands), since they need to gather operational experiences much earlier than customers, and they need to have a working backbone by the time customers start asking for IPv6.

- xDSL/L2 wholesale providers need to be ready for IPv6, as many of L3 ISPs depend on their services.
- Cheap/small routers for SO-HO/household needs to become IPv6 ready, however, price competition is extreme in the domain so IPv6 is tend to be moved to bottom of their priority list. Good thing is that the availability of IPv6 codebase from the KAME project would help them implement IPv6 into their products.
- There are multiple IPv6 protocol specification (not the basic ones) that needs to be finalized, or needs to be revisited/augmented. Also, vendors need to support those auxiliary specs.

We are now very optimistic about IPv6 deployment worldwide. Now the US government, Department of Defence, requires IPv6 support in their purchase requests, there are multiple European projects for IPv6 backbones, and there are a lot of Asian initiatives for IPv6 deployment. We Japanese needs to keep up the current pace, cooperate with other countries, and try to lead the IPv6 deployment worldwide.

5. Future plans

It is already decided that the KAME project will continue until spring 2006. As described above, we are still implementing and testing various internet protocol proposals at KAME, and try to keep *BSD IPv6/IPsec stack up-to-date. Also, our goal is now broadened not only to IPv6 but also to various internet technologies, such as DCCP, SCTP, multicast DNS and DNSSEC.

6. Conclusion

This paper talked about the history and status of the KAME project, some of the technical difficulties we have faced, and our future directions. For details of the KAME IPv6/IPsec implementation, you can just browse the source code at ftp://ftp.kame.net/pub/kame/snap/ (our code is heavily commented with references to RFCs), or refer to multiple papers presented in various occasions [Hagino, 2000; Tatuya, 1999] .

References

Bradner, 1993.
S. Bradner and A. Mankin, "IP: Next Generation (IPng) White Paper Solicitation" in *RFC1550* (December 1993). ftp://ftp.isi.edu/in-notes/rfc1550.txt.

Deering, 1995.
S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification" in *RFC1883* (December 1995). ftp://ftp.isi.edu/in-notes/rfc1883.txt.

Deering, 1998.
S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification" in *RFC2460* (December 1998). ftp://ftp.isi.edu/in-notes/rfc2460.txt.

Draves, 2003.
Richard Draves, "Default Address Selection for IPv6" in *RFC3484* (Feb 2003). ftp://ftp.isi.edu/in-notes/rfc3484.txt.

Hagino, 2000.
Jun-ichiro Hagino, "Mbuf issues in 4.4BSD IPv6/IPsec support (experiences from KAME IPv6/IPsec implemntation)" in *USENIX Freenix2000* (June 2000).

Hagino, 2000.
Jun-ichiro itojun Hagino, "Possible abuse against IPv6 transition technologies" in *draft-itojun-ipv6-transition-abuse-01.txt* (July 2000). work in progress material.

Hagino, 2003.
Jun-ichiro itojun Hagino and Craig Metz, "IPv4 mapped address on the wire considered harmful" in *draft-itojun-v6ops-v4mapped-harmful-02.txt* (Oct 2003). work in progress material.

Metz, 2003.
Craig Metz and Jun-ichiro itojun Hagino, "IPv4 mapped address API considered harmful" in *draft-cmetz-v6ops-v4mapped-api-harmful-01.txt* (Oct 2003). work in progress material.

Tatuya, 1999.
JINMEI Tatuya, YAMAMOTO Kazu, HAGINO Jun-ichiro, SUMIKAWA Munechika, INOUE Yoshinobu, SUGYO Kazushi, and SAKANE Shoichi, "An overview of the KAME network software: Design and implementation of the advanced internetworking platform" in *INET99* (June 1999).