Linux Is Now IPv6 Ready

Hideaki Yoshifuji
Keio University / USAGI/WIDE Project
yoshfuji@linux-ipv6.org

Abstract

Linux has included an IPv6 protocol stack for a long time. Its quality, however, was not quite good at the early stage. The USAGI Project was founded to promote this situation and provide high quality IPv6 stack for Linux systems. As a result of 5 years of our intensive activity, our stack is now certified as IPv6 Ready. It has been merged into main-line kernel so that the Linux IPv6 stack has enough quality to get the IPv6 Ready Logo now. To maintain the stack stable, we developed an automatic testing system, which greatly helps us saving our time. In this paper and the relevant presentation, we will show our efforts and technology to get the Logo and to maintain the quality of kernel. In addition, we will discuss our future plan.

1 Introduction

The current Internet has been running with the Internet Protocol Version 4, so called as IPv4, since the end of 1960s. At the end of 1980, the internet experts working at the IETF (Internet Engineering Task Force) recognized that we needed a new version of Internet Protocol to cope with rapid growth of the Internet. In 1992, this new version of protocol was named as IPng (IP next generation).

The first full-scale technical discussion on the IPng started in 1992 at the IETF. IPng, i.e., Internet Protocol Version 6 (IPv6), was intended to solve the various problems on the traditional IPv4, such as performance of packet forwarding, protocol extensibility, security and privacy.

According to the above principles, the basic specification of IPv6 was defined in 1994. After a series of experimental implementation and network operation (e.g., 6bone), the IPv6 technology is now getting into professional phase and applied to production. Commercial IPv6 services by Internet Service Providers and the applications running with IPv6 have been already available around us. This means that IPv6 stack implemented in any devices must be of production quality.

Linux system has also supported the IPv6 protocol as well as other operating systems such as FreeBSD, Sun Solaris and Microsoft Windows XP. Linux has included IPv6 stack since 1996 when early Linux 2.1.x version released. However, Linux IPv6 stack was not actively developed nor maintained for some time.

Considering above circumstances, USAGI Project was lunched in October, 2000. To deploy IPv6, it aims at providing improved IPv6 stack on Linux, which is one of the most popular open-source operating system in the world.

With a number of developments and treatments on problems, the quality has been remarkably improved. It is now good enough to be certified to the IPv6 Ready Logo Phase-1.

2 Quality Assessment of IPv6

There are some basic concepts about the quality assessment of IPv6. Among them, TAHI Conformance Test Suite, IPv6 Ready Logo Program Phase-1 and Phase-2 are most important ones.

2.1 TAHI Conformance Test Suite

TAHI Conformance Teat Suite is designed to examine the conformity to the IPv6 specifications. The details of the test are described in the test-scripts including the following fields; e.g.

- IPv6 Core
- ICMPv6
- Neighbor Discovery
- Stateless Address Autoconfiguration
- Path MTU Discovery
- Tunneling
- Robustness
- IPsec

This test suite is considered one of the de-facto standard tools for judgment of conformance of IPv6 stack. Linux IPv6 stack can also be examined by this suite.

2.2 IPv6 Ready Logo Program

IPv6 Ready Logo Program is a worldwide authorization activity for the interoperability on the IPv6. To obtain the certification, applicants should submit corresponding results of self test and pass the examination of the interoperability for the test scenario prior to the judgment. Some test sets, such as TAHI Conformance Test Tool that is a core part of TAHI Conformance Test Suite, are admitted as a tool for the Self Test.

2.2.1 Phase-1

The IPv6 Ready Logo Phase-1 indicates that the product includes IPv6 mandatory core protocols and can interoperate with any other IPv6 equipments. Self Test covers mandatory features of IPv6 core, ICMPv6, Neighbor Discovery, and Stateless Address Autoconfiguration. On the other hand, simple trial for the interoperability is carried out.

2.2.2 Phase-2

Phase-2 logo indicates that a product has successfully satisfied strong requirements stated by the IPv6 Logo Committee (v6LC). The v6LC defines the test profiles with associated requirements for specific functionalities.

The Core Protocols Logo covers the fields of IPv6 core, NDP, Addrconf, PMTU, ICMPv6, is designed to examine the MUST- and SHOULD- items in specifications, and its tests for interoperability are much more complicated than those of Phase-1. Other discussions are underway on the tests for IPsec, MLDv2, Mobile IPv6.

3 Quality Improvement Activities on Linux

As mentioned above, when the USAGI Project started, the quality of IPv6 stack is far beyond satisfaction though it was available on Linux. Linux IPv6 stack could not get good scores in the fields of Neighbor Discovery and Stateless Address Autoconfiguration in TAHI IPv6 Conformance Test Suite.

The members of USAGI Project and other contributers analyzed the problems of the stack, which are categorized as follows:

Improper State Transition. In Neighbor Discovery, improper state transition to the specification had been carried out. To solve the problem, the mutual dependency was sorted out in state machine to make the maintenance easier.

Inadequate time management. Time management at Neighbor Discovery and Stateless Address Auto-configuration did not have enough time accuracy. We conducted the structural reform mentioned above which enable the simplification of the management and more accurate time control.

Inadequate use of routing. In the previous method, there were some occasions where invalid route was used in an improper way.

Improper treatment against wrong input. Checks of input from outer sources were not adequate, improper treatments were going on for the wrong or malicious input.

The project results dealing with the above problems have been applied in main-line kernel step by step, until the version 2.6.11-rc2.

4 TAHI Automatic Running System

USAGI Project has been seeking for more featured and higher functional code, improving IPv6 stack for Linux, its libraries and applications. The results have been gradually accepted in the Linux community. For example, many improvements on IPv6 stack have been added in main-line kernels.

These good results are obtained by introducing TAHI Automatic Running System, which is improved system of TAHI Conformance Test Suite.

4.1 Background

The main objective of USAGI Project is to provide a better environment of IPv6 for Linux, and all the member of this project have been working hard for this purpose. One of those activities is to merge USAGI kernel patches to the main-line kernel.

Active improvement and amendment of the main-line kernel are under way on daily basis. As for the codes around the network, other patches as well as those by USAGI are tried to be taken in. Many maintainers and contributors are always wary of not being involved in mixing up bugs, but it is very difficult to avoid completely the possibility of regression after alternation.

The improved code is accepted widely in Linux community. While it is important to continue developing new functionality further more, to maintain the quality of present code is definitely necessary.

In order to solve these problems, a system was developed, which enables us to test the

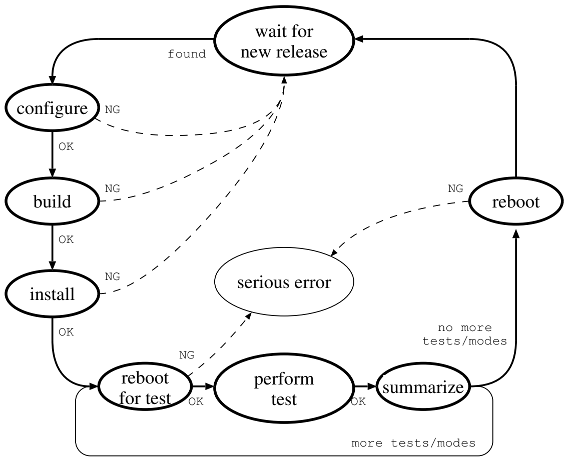


Figure 1: The Flowchart of the system

functions of IPv6 stack at each release of the snapshots of main-line kernel that is published every day. TAHI IPv6 Conformance Test Tool (`http://www.tahi.org`) provided by TAHI Project is used in order to test the functions of IPv6. Using this system that runs the test automatically, immediate amendment and tackling of the problems are possible even if some regression may be observed on the functions of IPv6. The system is open to the general public, and you can see it at `http://testlab.linux-ipv6.org` through the connect via IPv6.

4.2 Procedure of the System

The system is a bunch of some procedures, each of which consists of waiting for new release of the kernel, building-up, and testing. Those procedures are repeated, and the results are observed. The state transition of each procedure is shown in Figure 1.

The system waits for new kernel release when the test is not performed. The release objectives are not only stable version, but also rc version that is a preparing stage for stable version, together with bk version[1] that is released every night.

When the system finds a new release of a version, it begins to build up the kernel with automatic procedures of configuration, building-up, and installment. The logs in each procedure are preserved in the system, so that building errors can be analyzed. When the treatment of each procedure fails, the system assumes that the source contained the cause of problem, waiting for next release of version.

Once the building-up of the kernel has finished, the system carries out the test. This system is designed to run multiple tests with several settings for one kernel. For this purpose, NUT, the test target, is rebooted with proper mode such as router or host, and with proper settings such as IP address, prior to the launch of each test, and then the test is carried out. Each time when the test is finished, the result is shown in a table, which is compared with the previous records. That enable us to make sure if the regression might have occurred after the introduction of a new patch. The logs of each rebooting process and test result are preserved. If the system failed to reboot the target, it will stop its automatic operation and wait for manual resume after checking.

After the test of kernel, the system will be back to the stage of waiting for a new release of kernel. Before this transition to the waiting stage, the test target will be rebooted in order to get it back to the stable stage, with putting it back to a stable kernel verified. The logs at the stage of rebooting are also preserved.

4.3 Collected Data and Access to the Information

The system collects many kinds of data, such as the results of tests, the differences between

[1] as of April 2005

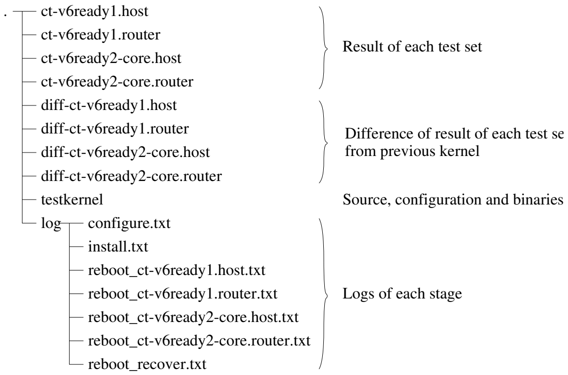

.
ct-v6ready1.host
ct-v6ready1.router
ct-v6ready2-core.host
ct-v6ready2-core.router
diff-ct-v6ready1.host
diff-ct-v6ready1.router
diff-ct-v6ready2-core.host
diff-ct-v6ready2-core.router
testkernel
log
configure.txt
install.txt
reboot_ct-v6ready1.host.txt
reboot_ct-v6ready1.router.txt
reboot_ct-v6ready2-core.host.txt
reboot_ct-v6ready2-core.router.txt
reboot_recover.txt
Result of each test set
Difference of result of each test se
from previous kernel
Source, configuration and binaries
Logs of each stage

Figure 2: Data Collected by the Autorun System

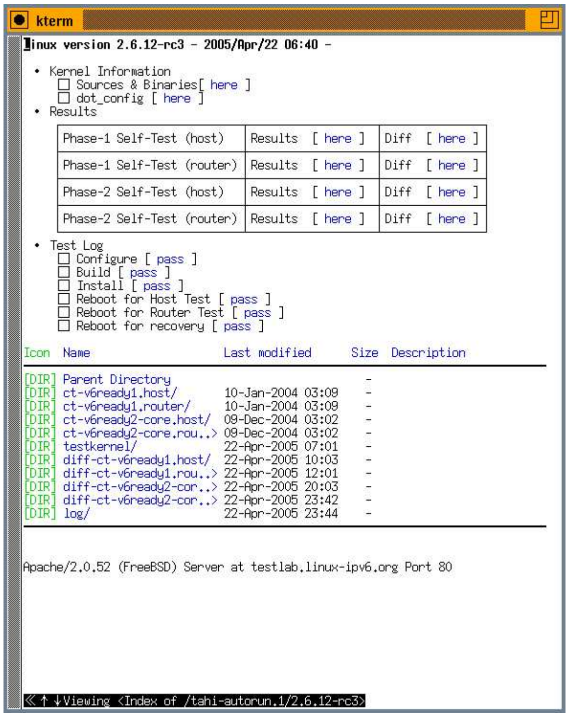
each test result and its previous equivalent test, the source and the compiled binaries, or various logs in each process of tests like kernel-build. Figure 2 shows what kinds of data should be collected at each release of kernel.

Each datum is exported by HTTP daemon and people can browse it using web browsers. The browser window is designed to seek the objective data open to the public as quick as possible. Figure 3 shows an access example through the web browser.

4.4 Development in Future

As of April in 2005, according to this system, IPv6 Ready Logo Phase 1 Self Test and Phase 2 Core Protocols Self Test are conducted only on the version 2.6 in main-line kernel. The expecting developments in future are as follows;

1. parallel proceedings to different kernel release, such as USAGI kernel and main-line kernel
2. supporting other tests, such as those for IPsec, MLDv2 and Mobile IPv6
3. General definition to the test target and test sets



```
Linux version 2.6.12-rc3 - 2005/Apr/22 06:40 -

  • Kernel Information
      □ Sources & Binaries[ here ]
      □ dot_config [ here ]
  • Results

      | Phase-1 Self-Test (host)   | Results  [ here ] | Diff  [ here ] |
      | Phase-1 Self-Test (router) | Results  [ here ] | Diff  [ here ] |
      | Phase-2 Self-Test (host)   | Results  [ here ] | Diff  [ here ] |
      | Phase-2 Self-Test (router) | Results  [ here ] | Diff  [ here ] |

  • Test Log
      □ Configure [ pass ]
      □ Build [ pass ]
      □ Install [ pass ]
      □ Reboot for Host Test [ pass ]
      □ Reboot for Router Test [ pass ]
      □ Reboot for recovery [ pass ]
Icon  Name                       Last modified      Size  Description
-----------------------------------------------------------------------
[DIR] Parent Directory                                -
[DIR] ct-v6ready1.host/          10-Jan-2004 03:09    -
[DIR] ct-v6ready1.router/        10-Jan-2004 03:09    -
[DIR] ct-v6ready2-core.host/     09-Dec-2004 03:02    -
[DIR] ct-v6ready2-core.rou..>    09-Dec-2004 03:02    -
[DIR] testkernel/                22-Apr-2005 07:01    -
[DIR] diff-ct-v6ready1.host/     22-Apr-2005 10:03    -
[DIR] diff-ct-v6ready1.rou..>    22-Apr-2005 12:01    -
[DIR] diff-ct-v6ready2-cor..>    22-Apr-2005 20:03    -
[DIR] diff-ct-v6ready2-cor..>    22-Apr-2005 23:42    -
[DIR] log/                       22-Apr-2005 23:44    -
-----------------------------------------------------------------------

Apache/2.0.52 (FreeBSD) Server at testlab.linux-ipv6.org Port 80

<< ↑ ↓Viewing <Index of /tahi-autorun.1/2.6.12-rc3>
```

Figure 3: Access Example through Web Browser

These items are now being discussed under development.

5 Linux Is IPv6 Ready

As described in Section 2.2, IPv6 Ready Logo Program is an international activity for proof of interoperability. As of December 2004, more than 120 products have been approved with gaining the Phase-1 certification, which means those products have the basic interoperability in IPv6.

IPv6 used to be classified as EXPERIMENTAL in Linux, so that people are worrying for long time that IPv6 in the Linux would not be useful. Getting this logo, however, the anxiety would be expelled completely.

5.1 USAGI Project

USAGI Project decided to join this program to gain the international certificate, which will show what credible results we are offering.

In 2004, USAGI Project took part in IPv6 Ready Logo Program with its patched kernel and tool. In February, the Project obtained IPv6 Ready Logo Phase-1 on both functions of host and router with its product based on 2.6 kernel and with its enhanced tool. In September 2004, and in April 2005, it gained IPv6 Ready Logo Phase-1 on 2.4-based kernel, in addition to 2.6-based one, on the functions of host and router.

5.2 Main-line

Many improvements by USAGI Project members and other developers were unified into the main-line kernel in the 2.6.11 timeframe, and the version 2.6.11-rc2, with patched radvd (router advertisement daemon), is finally approved with IPv6 Ready Logo Phase-1.

In addition to this, the version 2.6.12 will include the kernel function that is needed to get IPv6 Ready Logo Phase-2 certificate.

5.3 KNOPPIX/IPv6

USAGI Project collaborated with KNOPPIX/IPv6 (`http://www.alpha.co.jp/knoppix/ipv6/`), which uses the provided code by USAGI to their products, and helped them to take part in IPv6 Ready Logo Program. KNOPPIX (`http://www.knopper.net/knoppix/`) is one of the major Linux distributions which makes it possible to boot with single CD without any special installing operation. USAGI Project collaborated with AIST (National Institute of Advanced Industrial Science and Technology), Alpha Systemss Inc. to develop special IPv6-aware KNOPPIX based on KNOPPIX Japanese Edition (`http://unit.aist.go.jp/itri/knoppix/`). The code provided by USAGI Project was integrated, and the resulting product was named "KNOPPIX/IPv6" (`http://www.alpha.co.jp/knoppix/ipv6/`). With KNOPPIX/IPv6, together with the merit of KNOPPIX that "only starting of CD-ROM is needed without installment and setting-up" and the high quality IPv6 protocol stack by USAGI Project, a new technical 1 CD OS world has been achieved where beginners can easily experience the IPv6 world.

To summarize the dealing with situation of KNOPPIX/IPv6, major desktop applications such as web browsers (Mozilla, Konqueror), mail clients (Sylpheed, Kmail) support IPv6. On the other hand, as for coping with fundamental IPv6 networking, 6to4 is adopted. This function is supported, because users do not always connect their machines to the global IPv6 Internet. Even if a user connects to the IPv4-only network, KNOPPIX/IPv6 automatically detects it and configures 6to4 tunnel to the outside local network. Therefore, from now on, it is possible for users to enjoy more sophisticated network without realizing whether it is IPv4 or IPv6.

In September 2004, KNOPPIX/IPv6 could obtain IPv6 Ready Logo Phase-1 on both kernel versions based on 2.4 and 2.6. The necessary tests to confirm interoperability were conducted at laboratory of USAGI Project in Keio University as a cooperative work of the Project and developers of the KNOPPIX Japanese Edition.

5.4 Development in Future

USAGI Project is going to participate in IPv6 Ready Logo Phase-1 Program to improve the quality of interoperability.

On the other hand, in the new IPv6 Ready Logo Phase-2 Program, the aim of which is verification of the system whether it is available in the real network environment, not only basic IPv6 functions but also IPsec, MIPv6, and MLD are subject to verification. USAGI Project is going to participate actively in the Phase-2 Program, and will play an initiative role in the quality improvement.

The Self Test for IPv6 Ready Logo contains a lot of its functions, playing a great role for the quality improvement and maintenance of the Linux, contributing to the less personal burden of labor. However, it does not warrant liability against the stress and attack from the outside, nor the stability in SMP (symmetric multiprocessing). Now, IPv6 is enabled by default in Linux, it is more important to maintain the stability of the system on the more complicated in higher stage. The members of this project will continue to achieve this objective through fulfillment of the experiments and the practical environment.

Acknowledgements

This work was performed based on the USAGI Project and the author is grateful to valuable comments and helpful discussion with Prof. Jun Murai at Keio University and Prof. Hiroshi Esaki at The University of Tokyo.

Proceedings of the
Linux Symposium

Volume Two

July 20nd–23th, 2005
Ottawa, Ontario
Canada

Conference Organizers

Andrew J. Hutton, *Steamballoon, Inc.*
C. Craig Ross, *Linux Symposium*
Stephanie Donovan, *Linux Symposium*

Review Committee

Gerrit Huizenga, *IBM*
Matthew Wilcox, *HP*
Dirk Hohndel, *Intel*
Val Henson, *Sun Microsystems*
Jamal Hadi Salimi, *Znyx*
Matt Domsch, *Dell*
Andrew Hutton, *Steamballoon, Inc.*

Proceedings Formatting Team

John W. Lockhart, *Red Hat, Inc.*