Testing IPv6 implementations with TAHI

TIPSTER6 project

Abstract

IPv6 has now reached the level of being suitable for enterprise services. Several vendors (e.g. Cisco, Compaq, IBM, and Sun) or organizations (Linux, FreeBSD, OpenBSD, NetBSD) are already shipping their products with IPv6 support. It is expected, that most of the vendor will follow the trends, and expected to ship their IPv6 ready products by the end of this year (e.g. Windows XP by Microsoft).

When introducing new technology, it is important to find out the products' compliance to the standards, and their interoperability. However, the new Internet devices (handheld, game devices, home-appliances) can generate a further challenge.

This paper describes the current state of the ongoing interoperability and compatibility test of IPv6 implementations in TIPSTER6 project. The TIPSTER6 project is a follow-up project of the IPv6 testing and operating experience done at two department of Budapest University of Technology and Economics (Centre of Information Technology, Department of Control Engineering).

The results are practically an extension of the previous report appeared in the TERENA Networking conference in 1998. [TNC98]

Introduction

The TIPSTER6 project is formed from Centre of Information Technology of Budapest University of Technology and Economics Budapest, Department of Control Engineering and Information Technology of Budapest University of Technology and Economics, KFKI Research Institute for Particle and Nuclear Physics, and MATÁV PKI to facilitate the IPv6 deployment in Hungary especially in the HUNGARNET. See more http://tipster6.ik.bme.hu/tipster6_en.html , The project members have more than 4 years experience in the IPv6.

Recently several IPv6 implementations have been made public. The TIPSTER6 project is concentrated on checking the implementations' adherence to standard, their basic IPv6 functions and their interoperability. The TIPSTER6 selected two tools for testing IPv6 conformance: the TAHI conformance test suite [TAHI], and the Sun Packet Shell [SUNPKT]. The selection is based on the availability and coverage. Both test suite has to be extended to able test some feature of the IPv6 or Implementations. In this paper we are concentrating on the TAHI test results. We have used the results of the extended Sun Packet Shell for checking purpose.

We are continuing the testing of the different implementations with focus on other areas like interoperability and performance.

The Tested Systems

Majority of tested systems are already released products, so when we found some defects we tried to contact the vendor or the support forum of the particular product to resolve the problem or to provide feedback for the product improvement.

- Pentium III with FreeBSD 4.2 and FreeBSD 4.3
- Sparc Ultra 5 with Solaris 8
- Pentium III with Linux 2.2.19 and Linux 2.4.3
- RS6000-170 with AIX 4.3.3

The summary of the results has already presented in GEANT project TF-NGN working group in Tromsø.

Overview of implementations and their test results

FreeBSD 4.2 and FreeBSD 4.3

All of the BSD implementations are based on the KAME implementation [KAME], that is considered one of the most feature-full and most stable implementation.
To able to test the FreeBSD under TAHI it was necessary to configure IPv6 from the `/stand/sysinstall`. According to the TAHI documentation it was recommended to remove the `/etc/resolv.conf` to disable DNS address resolution and also recommended to remove the IPv4 default route from the `/etc/rc.conf`.
In the case of FreeBSD 4.2 the IPv6 , the ICMPv6 and the robustness test ran correctly with exception of one test: Routing header handling at the end node. Examining the results with tcpdump we found that the tested machine reacted correctly to the testing sequence in that particular test-case, but it sent output to a different interface. We found that KAME stack is susceptible to misconfiguration if more that one network interface exist in the tested node and used it is in stateless autoconfiguration mode. This problem cannot be considered to be a bug, since the network node cannot decide itself which interface is the default configuration interface if more than one exist. That is why an option is available for KAME implementations to define the default network interface. In any case it worth to warn the users about this feature. This problem in the test with FreeBSD 4.3 (with correct setup) disappeared.
Because of the misconfiguration, during of testing Neighbour Discovery these problem with FreeBSD 4.2 persisted: problem with Router Solicitation messages, problem of Redirect and problem of Router Advertisement. In FreeBSD 4.3 some of these errors has been resolved: correction of the router/neighbour-discovery cache handling when incorrect Neighbour Advertisement packets are arriving.
The PATH MTU discovery is also has problem in FreeBSD 4.2 that mostly solved in FreeBSD 4.3.
Unfortunately the Stateless Address Autoconfiguration is only working at the basic level. The more complicated autoconfiguration cases are became failed. The most important problem existed in the Duplicate Address Detection, that is persisted in the case of FreeBSD 4.3. This might need some more investigation, since the test results published by the TAHI are virtually errorless in the case of Address Autoconfiguration.
The 2 tunnel tests have failed and 2 tunnel tests have succeed, that is reasonable since the KAME IPv6 stack implement only the configured tunnels, but not the automatic tunnels.

	Total	OK	WARN	FAIL	Overall
IPv6 spec	37	36/37	0/0	1/0	~/✓
ICMPv6	16	16/16	0/0	0/0	✓/✓
ND	58	31/53	1/2	26/3	✗/~
Stateless AC	56/57	28/31	7/9	21/17	✗/✗
PMTU	4	0/3	0/1	4/0	✗/✓
Tunnel	4	0/2	0/0	4/2	✗/✓
Robustness	4	4/4	0/0	0/0	✓/✓
Summary					~/✓

Summary of FreeBSD TAHI results

Recommendation for FreeBSD users: You can use the FreeBSD for IPv6 services since it is mature and conforming the standards. You can follow each important changes if you “stay stable with FreeBSD”. If you need some fancy feature like NAT-PT, Mobile-IPv6 you can use KAME development stack that is high quality, but sometimes is not operational due to the ongoing developement.

AIX 4.3.3

The AIX implementation is based on the INRIA IPv6 implementation [INRIA], that was one of the most mature and stable implementation in 1998-1999.
To able to test the AIX under TAHI it was necessary to extend the TAHI suite with the necessary command and test sequences, and then configure IPv6 from the `smit` (the graphical management console). According to the TAHI documentation it was recommended to remove the `/etc/resolv.conf` to disable DNS address resolution and also recommended to remove the IPv4 default route.
In the case AIX the IPv6 standard compliance is rather good but there are some problems: the erroneous Next Header incorrectly reported back to the sender node, the Solicited Node Multicast Address not handled perfectly in each cases, not handling correctly the 65536 octet length packets and incorrectly ordered fragments are handling has some deficiency.
The ICMPv6 compliance is not perfect: most of the errors are related to the link-local-address handling. However there are problems with the ICMPv6, the basic ICMPv6, like ICMPv6 echo, destination unreachable, and ICMPv6 error reporting working correctly.
The IPv6 robustness tests ran without any error.
During testing the Neighbour Discovery, we found a lot of problem in AIX: problem with Router Solicitation messages, problem of Redirect and problem of Router Advertisement. The biggest problem is that, AIX is sometimes does not send Neighbour Solicitation messages, it simply fills out the neighbour cache from the incoming packets. Handling the neighbour discovery messages so roughly is unexpected, since according to our previous test the INRIA stack was quite good at handling them.
The PATH MTU discovery also has problems in AIX 4.3.3. Only the initialisation is working.
Unfortunately the Stateless Address Autoconfiguration is also only working at the basic level. The more complicated autoconfiguration cases are became failed.
The 3 tunnel tests failed failed and 1 tunnel tests has succeed. This is bug of our TAHI extension since both configured tunnels, and automatic tunnels working correctly.

	Total	OK	WARN	FAIL	Overall
IPv6 spec	37	30	0	7	✓
ICMPv6	16	8	0	8	~
ND	58	10	0	48	×
Stateless AC	57	2	18	37	×
PMTU	4	1	0	3	×
Tunnel	4	1	0	3	×
Robustness	4	4	0	0	✓
Summary					~

Summary of AIX TAHI results

Recommendation for AIX users: You can use the AIX for IPv6 services since it is stable and mostly conforming the standards.

Solaris 8

The Solaris has its own implementation is based on the standards. The test was run on two different kernel version: the kernel patch version 108528-3 and the kernel patch version 108528-6, but the later is summarised only the table.
To able to test the Solaris under TAHI it was necessary to configure the IPv6 at the installation. According to the TAHI documentation it was recommended to remove the `/etc/resolv.conf` to disable DNS address resolution and also recommended to remove the IPv4 default route from the `/etc/defaultrouter`.
In the case Solaris 8 the IPv6 , the ICMPv6 and the robustness test ran correctly.

Testing the Neighbour Discovery on Solaris 8 showed us, that there are some problem: Neighbour solicitation messages are not sent several times, receiving Neighbour Solicitation and Router Solicitation is sometimes problematical especially in more complicated cases. The problems existed in handling Router Advertisement messages almost completely corrected in the kernel patch version 108528-06. The newer version of kernel also improves the Redirect message handling.

Unfortunately, the PATH MTU discovery, the Stateless Address Autoconfiguration and the tunnelling worked only at the basic level.

The results are bit unexpected, since we have been using Solaris 8 with IPv6 for a long time with no serious problem. We think we has to retest Solaris in more controlled environment, to be able to distinguish the real problems, from the measurement error.

We wanted to test the IPSec on Solaris, but unfortunately Solaris 8 IPSec does not support asymmetrical ICMP IPSec policy.

	Total	OK	WARN	FAIL	Overall
IPv6 spec	37	37	0	0	✓
ICMPv6	16	16	0	0	✓
ND	58	37	3	18	~
Stateless AC	56	1	14	41	×
PMTU	4	1	0	3	×
Tunnel	4	3	0	1	~
Robustness	4	4	0	0	✓
Summary					~

Summary of Solaris TAHI results

Recommendation for Solaris users: You can use the Solaris for IPv6 services since it is stable and conforming the standards. You should patch your system at latest kernel patch level (e.g. `108528-08 for sparc`, or `108529-08 for x86`)

Linux 2.2.19 and Linux 2.4.3

Linux has its own implementation based on IPv6 RFCs.

To able to test the Linux under TAHI it was necessary to install and configure the IPv6. For testing purpose we used Debian. We have removed a default route from `/etc/network/interfaces` file and switched of DNS resolver removing `/etc/resolv.conf`.

We have tested both kernel with 2 different settings:

default settings (noted with no extension):

```
net.IPv6.conf.all.autoconf = 1
net.IPv6.conf.all.accept_ra = 1
net.IPv6.conf.all.accept_redirects = 1
net.IPv6.conf.all.forwarding = 0
net.IPv6.conf.all.router_solicitations = 3
```

and recommended settings (noted with –2 extension):

```
net.IPv6.conf.all.autoconf = 0
net.IPv6.conf.all.accept_ra = 0
net.IPv6.conf.all.accept_redirects = 0
net.IPv6.conf.all.forwarding = 1
net.IPv6.conf.all.router_solicitations = 0
```

In the case of Linux we found a serious bug in the kernel, more exactly in the IPv6 fragment handling that make Linux kernel panic. We informed the developers via mailing list, and the fix is available in the Linux Kernel 2.4.4.
The IPv6, and the ICMPv6 test provided the best result with 2.4.3 version kernel + recommended settings. Using these setting only few errors remained related to multicasting and jumbograms.
Testing the Neighbour Discovery, we found that the 2.4.3 kernel is better than 2.2.19, but it still contains lots of bug.
The PATH MTU discovery worked only at the basic level..
Unfortunately the Stateless Address Autoconfiguration hardly work at all..
The tunnel tests have succeed, and also all the robustness tests have passed.

	Total	OK	WARN	FAIL	Overall
IPv6 spec	37	26/33	0/0	11/4	~/✓
ICMPv6	16	4/15	0/0	12/1	×/✓
ND	58	6/8	0/2	52/48	×/×
Stateless AC	54	1/1	8/12	45/41	×/×
PMTU	4/2(rest is failed)	0/1	0/0	4/1	×/~
Tunnel	4	1/4	0/0	3/0	×/✓
Robustness	4	0/4	0/0	4/0	×/✓
Summary					×/~

Summary of Linux TAHI results

Recommendation for Linux users: You can use the Linux for IPv6 services, but you should use at least the kernel version 2.4.4 since it is more stable and more adherent to the standards. You can consider using the USAGI project patch for Linux [USAGI], since according to their TAHI test results , it seems to solve most of the problem.

Summary

The IPv6 protocol is mature and standardised in RFCs. Some auxiliary standards are under development. The operating system vendors finally started to support IPv6 in their products.
We can safely claim that all the tested operating systems work well at the basic level, but there are some minor implementation problems.
Based on our tests all the four implementation can be used for providing IPv6 services, if recommendations are followed.
There are far greater differences when we are looking to the applications. All platforms have some basic tools, but to provide service, you need a more broader application support. We focused our activities to examine the applications IPv6 support.
We are also planning to test some important systems like: Tru64 UNIX, Window 2000 and HP-UX.
The detail of the tests are available in the TIPSTER6 project webpage [TAHI_TESTS].

References

[TAHI] TAHI Project is the joint effort formed by the three Japanese organizations with the objective of developing and providing the verification technology for IPv6. http://www.tahi.org
[KAME] KAME Project is a joint effort of seven companies in Japan
to provide a free IPv6 and IPsec (for both IPv4 and IPv6) stack for BSD variants to the world. http://www.kame.net

[TNC98] J.Mohácsi, Sz. Szigeti: Testing IPv6 implementations, TERENA Networking conference, 1998, Oct, Dresden. In English http://tipster6.ik.bme.hu

[USAGI] USAGI (UniverSAl playGround for Ipv6) Project works to deliver the production quality IPv6 protocol stack for the Linux system. http://www.linux-ipv6.org

[TAHI_TESTS] Tahi test results in TIPSTER6 project. http://tipster6.ik.bme.hu/tahi_tests/