Testing IPv6 implementations

János Mohácsi,Szabolcs Szigeti,Tamás Máray

Department of Control Engineering and Information Technology
Technical University of Budapest,
Mûegyetem rakpart 9., H-1111, Budapest

e-mail:{mohacsi,pink,maray}@fsz.bme.hu

Contact: Szabolcs Szigeti (pink@fsz.bme.hu)

Tel: +36-1-4632649
Fax: +36-1-4632204

Abstract

The development of IPv6 - the next generation protocol to step in place of the current Internet Protocol - has reached the level of being suitable for practical usage. Several vendors are already shipping their products with IPv6 support. It is expected, that every major hardware and software vendor will step ahead with their own implementation in one year time.

With the introduction of a new technology, it is crucial to find out the implementations adherence to the standards, and their interoperability.

This paper describes the current state of the on-going interoperability and compatibility test of IPv6 implementations. These tests were performed at the Department of Control Engineering and Information Technology, Technical University of Budapest.

We have examined the IBM Aix, Digital Unix, FreeBSD/INRIA, FreeBSD/WIDE, Linux and FTP Software Secure Client implementations.
The following two implementations were issued recently thus we have not yet performed all the tests on them: FreeBSD KAME, Microsoft Research IPv6 for Windows NT.

Keywords: Network technology, Protocol, Internet, TCP/IP, IPv6

Table of Contents

1. Introduction

Recently several IPv6 implementations have been made public. The Computer Network Research Group of the Department of Control Engineering and Information Technology, Technical University of Budapest, with its more than one and a half years of IPv6 experience, have tested a number of host implementations. The test were concentrated on checking the implementations adherence to standard, their basic IPv6 functions and their interoperability [7].

We have deliberately not tested performance, since the different hardware platforms would have rendered these results meaningless.

It is important to note, that these experiments indicate the state of the tested systems at the time of writing. All of the implementations are constantly improving. We are planning to test other implementations (host and router) in the near future.

2. The tested systems

- Pentium PC /486 PC, FreeBSD 2.2.5 + INRIA implementation; (1998 January snapshot).[8]
- Pentium PC /486 PC, FreeBSD 2.2.5 + WIDE implementation; (19980209 snapshot).[9]
- 486 PC, Linux 2.1.85 + inet6-apps0.26 +net-tools-980126.[10]
- Alphaserver 2000, Digital UNIX 4.0B + IPv6 v.6.112.[12]
- RS6000/580, AIX 4.3.[11]
- 486 PC, Windows 95, FTP Software's Secure Client 3.0 implementation.[17]
- Pentium PC, FreeBSD-2.2.6 + KAME-980531-fbsd226-stable.[15]
- Pentium PC, Windows NT 4.0 + Microsoft Research IPv6.[16]

Besides these, all of the major systems have or will have shortly experimental IPv6 implementation. (Sun Microsystems: Solaris, Hewlett Packard: HP-UX, Silicon-Graphics: IRIX, DEC: OpenVMS, SCO: Gemini, Novell: Netware, Mentat: MacOS)

3. Services Provided by the IPv6 implementations

The next table (Table 1.) is based on the documentation and descriptions of the different systems. The results of our test is described in the section of Test Results.

Implementation / Services	FreeBSD INRIA	FreeBSD WIDE	Linux	Digital UNIX	AIX 4.3	FTP-win95	FreeBSD KAME	Microsoft IPv6
Stateless autoconfiguration	+	+	+	+	+	+	+	+
Neighbour discovery	+	+	+	+	+	+	+	+
Routing Redirection	+	+	+	+	+	?	+	?
RFC 1897 type address support[6]	+	+	+	+	+	+	+	?
EUI-64 address support[6]	+	+	+	+	+	-	+	+
Multihoming	+	+	+	+	+	?	+	?
Tunneling Configured, Automatic	+	+	+	+	+	-	+	+
Router mode	+	+	+	+	-	-	+	-
Using DNS service [1]	+	+	+	+	+	-	+	-
DNS support [2]	+	-	+	+	+	-	+	-
Supported Media								
Ethernet	+	+	+	+	+	+	+	+
Tokenring	-	-	-	-	+	-	-	-
FDDI	+	+	?	+	+	-	+	-
ATM	+	+	+	-	-	-	+	-
PPP	+	+	?	-	-	-	+	-
RFC2133 application programming interface [4]	+	+	+	+	+	-	+	?
Applications								
telnet	+	+	+	+	+	-	+	-
ftp	+	+	+	+	+	-	+	-
mail [3,4]	+	-	+	-	-	-	+	-
http server [4]	+	-	+	-	-	-	+	-
http client [4]	+	-	+	-	-	-	+	-
NFS	+	-	-	-	-	-	-	-
X11R6	+	-	-	-	-	-	+[6]	-
Authentication and Encryption								
Authentication	+	+	+	-	+	+	+	-
Encryption	-[5]	+	+	-	+	+	+	-

Legend:

1. Currently, in all of the implementation the resolver library works over IPv4 only.
2. In every Unix implementation, Berkeley DNS version 4.9.6 or 8.1.1 can be used. It has AAAA

resource record support. The systems indicated contain it by default [5].
3. Modified sendmail is part of the distribution.
4. Can be ported to any RFC 2133 compliant system.
5. The original INRIA implementation contains the code, but cannot be exported due to French law.
6. Available but not tested by KAME.

- **+** According the documentation service is included in the package.
- **-** According the documentation service is not included in the package.
- **?** The presence of the feature is not indicated in the documentation

Table 1.: Service provided by the IPv6 implementations

4. Overview of the implementations

Among the freely distributable IPv6 implementations the WIDE and the Linux implementation are very similar to each other. Both are based on the implementation and concept of the Naval Research Laboratory (NRL) of USA. Principal element of their implementation is that there are independent TCP-UDP layers for IPv6 and IPv4. This is shown very clearly in the WIDE implementation, where there is a separate inetd6 daemon for launching the IPv6 network services. The other characteristic element of these systems is that autconfiguration messages are processed by the kernel without any help of user processes.

The other reference implementation is made by INRIA (France). Its main feature is that IPv6 and IPv4 share the common TCP-UDP transport layer. The autoconfiguration protocol messages are processed partly by a user process. The IPv6 support of IBM's AIX is based on the INRIA code and Silicon Graphics has chosen it as a base for the IRIX IPv6 code. Digital's own implementation also shows some similarity to INRIA.

In the NRL based implementation IPv6 is regarded as ``yet another protocol''. The advantage of the separated protocol stack is, that it is easier to develop, maintain and test the new protocol, without disturbing the working IPv4. But this concept leads to some inefficiency, since both protocol stack independently allocates its buffers. On the other hand, the INRIA implementation regards to IPv6 as a new protocol that will eventually replace IPv4. However development of this integrated protocol stack is more difficult, while its memory handling can be more efficient.

The KAME implementation is a rather new system. It is largely based on the WIDE IPv6 code. It also contains number of enhancements and large amount ported application.

The Microsoft Research IPv6 implementation is an early release of IPv6 for Windows NT.

5. Ease of installation of IPv6 implementation

Ease of installation of the tested IPv6 implementations ranges from perfect to almost unusable.

5.1 FreeBSD INRIA implementation (Snapshot 1998 January)

The INRIA implementation can be installed fairly easily. The IPv6 distribution package should be extracted on top of the FreeBSD 2.2.5 source tree, then everything has to be recompiled. After installing a new IPv6 kernel, every necessary package should be installed. The system is then ready for IPv6 configuration and testing. Even though that the extension package can be recompiled very easily, it takes quite long time (approximately 2 hours on a 133 Mhz Pentium). The new and the rewritten programs are placed into their usual location, so this package can be considered as a production

version, not a test system. Further disadvantage of the INRIA package is that it is not available in patch or CVS update. About the installation there is short, but enough information available as part of the distribution.

5.2 FreeBSD WIDE implementation (Snapshot 19980209)

Installation of the WIDE package is more difficult. The new version contains the source of the new and the modified programs, a library (libinet6) for them and 2 patch files (for the kernel, and the include files). About the installation a short, comprehensive documentation is provided in English and in Japanese. Reconfiguring the kernel is not difficult, but there is an annoying dependency between IPv6 and IPSEC: a new IPv6 kernel cannot be built without IPSEC. However compiling the utilities and the different tools are not easy. Sometimes the kernel and the include files have to be checked since the programs use undefined constants or structures that can be resolved only by looking at the kernel source and include files. The new IPv6 tools are installed in the /usr/local/v6 directory. We can say that the WIDE package is more of an experimental system than a really usable implementation.

5.3 Linux Implementation + inet6-apps0.26 + net-tools-980126

Installing the Linux IPv6 packages are quite difficult. The kernel and the related tools have to be obtained from different sources. Only limited information is available about the installation. The IPv6 code is included in every developer kernel (2.1.x). Creating an IPv6 compliant kernel is quite complicated with the make config. The configurator asks thousands of options that are not documented. Any mistake will result in an unstable or not working kernel. The new IPv6 tools are split into two packages. One would expect that first the basic components from the net-tools package (ifconfig, netstat etc.) have to be installed. However first the applications (like ftp) from the inet6-apps should be installed because this package contains the libinet6 library (usual on NRL implementation) which contains the support of IPv6. Even after finding out this, the installation is not easy because the undefined structures and constants should be searched for in the kernel and header files. Another problem is that make doesn't stop at errors, but ignores them. The IPv6 compatible programs, libraries are installed into the /usr/inet6 directory. The Linux IPv6 package is far from stable release, it is rather a test system.

5.4 Digital UNIX 4.0B + IPv6 v.6.112

The Early Adopter's Kit of Digital is easily installed. After unpacking the package the binary and documentation kits are installed with setld. After this a new kernel must be compiled. The whole installation process is well documented. The new utilities are installed under the /usr/opt/IP6112 directory. From the standard place symbolical links point to the modified tools. Although DEC doesn't provide support for this kit, reliability, and stability makes it generally usable system. There are two minor problems to be solved: the ATM and IPv6 cannot be used together. IPv6 is not supported on SMP systems.

5.5 AIX4.3

The IPv6 is integrated into the AIX 4.3 operating system, therefore after installing this, our RS6000 immediately supports IPv6. There are two small problems with this implementation: absence of real documentation for the whole AIX system. The authentication and security package is not part of the AIX operating system, it is only available in the Bonus Pack.

5.6 FTP Software Secure Client 3.0

This is the only Windows 95 IPv6 package among the examined implementations. As usual in the Windows environment, it can be installed very straightforward way. Even though the installer of the

package warns that the TCP/IP stack of Windows will be replaced, Windows hangs if TCP/IP is used during the installation. In order to resolve this problem the installer of Secure Client should be run again or first the Microsoft TCP/IP stack should be removed.

5.7 FreeBSD KAME

The KAME implementation can be installed quite easily. The system contains similar components to the WIDE implementation: source of the new and modified programs, a library (libinet6) and the 2 patches. There are also some ports that are new compared to the WIDE implementation. For installation a good documentation provided in English. Compiling the ipv6 compatible kernel is as easy as previously in the WIDE. The tools and utilities can be compiled quite cleanly, with some warnings. The new ipv6 compatible commands and manual pages are installed in the /usr/local/v6. The most attractive feature of the KAME is the ports of different networking applications: e.g. Mozilla.

5.8 Microsoft Research ipv6

This IPv6 package installs without any problems, using the standard method (via Control Panel / Network) in Windows NT. It may be uninstalled in a similar fashion. This implementation is available for Windows NT 4.0 and 5.0 (beta), in binary and source distribution.

6. Configuration

6.1 FreeBSD INRIA

The INRIA implementation can be configured fairly easily, although it is not documented very well. Using the example configuration supplied with the system the host configuration can be carried out quickly. Creating the router configuration and establishing tunnels are not so easy. The configuration would be much easier if there were a topology description supplied with the example. The commands are well documented.

6.2 FreeBSD WIDE

Configuration of the WIDE implementation is moderately difficult. There is a short manual about the new commands, but a real description is missing. Creating a host configuration is very easy by following the example. Setting up a router, similarly to the INRIA implementation, is quite hard. One has to look at the source of the program to find out the usage of the different switches. Another interesting feature of the system is that a daemon has to run to use the automatic tunnels.

6.3 Linux

Configuration of the Linux is fairly easy. There is a small "how-to" about the commands, but no real description is available. After reading the "how-to", creating a host configuration is very easy. Making a router configuration is more difficult. The source of the commands has to be read for understanding the program options. About the router advertisement daemon a short description is given which helps. The configuration of the tunnels is not perfect, since the automatic and configured tunnels are not separated enough.

6.4 Digital

The Early Adopter's Kit of the Digital is easily configured, since a very comprehensive guide is supplied with the package. This guide examines thoroughly the different configurations. Even the router configuration is simple, because the guide describes it in depth. The only missing piece of the documentation is the manual pages of the new commands which are not ready yet. The supplied

ipv6_setup shell script generates a nearly perfect configuration, any problem can be determined and solved with the help of the guide.

6.5 AIX4.3

The IPv6 is integrated into AIX 4.3 very well. Every option can be changed from SMIT. Creating a host configuration is no more than a few clicks in SMIT. Setting up tunnels are also simple. Router configuration is not available since AIX version 4.3, issued in 1997 November, doesn't support router configuration. The only missing thing is the documentation.

6.6 FTP Software Secure Client 3.0

Configuring the FTP Software product only requires switching on one option, and IPv6 is "running". More precisely it would work, only if it had support for not only the Provider Based Unicast Addresses. The 6bone and the whole IPv6 community has moved from this less structured addressing method to the new Globally Aggregatable Unicast Addresses in November 1997. Except for the Secure Client all the examined systems support this new standard. The Secure Client does support neither tunnels nor router mode.

6.7 FreeBSD KAME

The configuration of the KAME is better than of the WIDE. Configuring the KAME is easier than before in the WIDE because there are manual pages in addition to the short description.

6.8 Microsoft Research IPv6

The Microsoft IPv6 implementation is very basic, thus configuration is not flexible or user-friendly.
Interface address configuration is only available as automatic configuration, to configure a global address one needs a router or DHCP server to supply autoconfiguration information.
Configured tunnels can only be set up by editing the registry.
GUI configuration tools are promised to be in forthcoming releases.

7. Documentation

7.1 FreeBSD INRIA

The INRIA package documented moderately well. The new utilities have manual pages and they are updated regularly. At present some documentation in French is more complete than in English. A comprehensive guide, like the one in the Digital package, is not available.

7.2 FreeBSD WIDE

The WIDE kit is documented poorly. There are two small text files about the installation and commands in English. A few of the programs are documented in Japanese.

7.3 Linux

The documentation of the Linux package is poor as well. There are some "How-To-IPv6", but they are contradictory and do not provide much help. Except from these everything can only be discovered by reading the source of the programs.

7.4 Digital

Documentation of the DEC system is exemplary. The installation, configuration and programming guides are available either in HTML, or in Postscript. The examples of the guides are available too. Unfortunately the updated manual pages are missing.

7.5 AIX 4.3

Currently the on-line documentation is not part of AIX 4.3. At the time of issuing the AIX 4.3 the documentation was only in beta, - since IBM is moving to HTML based documentation -, thus almost nothing was available about AIX and IPv6. From the IBM web site the full documentation can be downloaded and hopefully it will also be included in the next release of AIX.

7.6 FTP Software Secure client 3.0

For this program FTP software provides a comprehensive manual, that doesn't provide any useful information about IPv6.

7.7 FreeBSD KAME

The KAME kit is documented better than the WIDE, but the manual pages are still quite short.

7.8 Microsoft Research IPv6

A small readme file is supplied with their package, which gives basic information on the use and configuration of the limited capabilities of this implementation.

8. Description of the tests

The tests were performed using the experimental IPv6 network in our lab. Some of the test methods are based on the JOIN tests [14], but this paper provides complementary and additional information about the current status of IPv6 implementation. The figure (Fig. 1) below describes the configuration of the network. The network is connected to the 6bone through a FreeBSD/INRIA machine, which also acts as a router [13].

We monitored the Ethernet network traffic using the tcpdump utility [3].

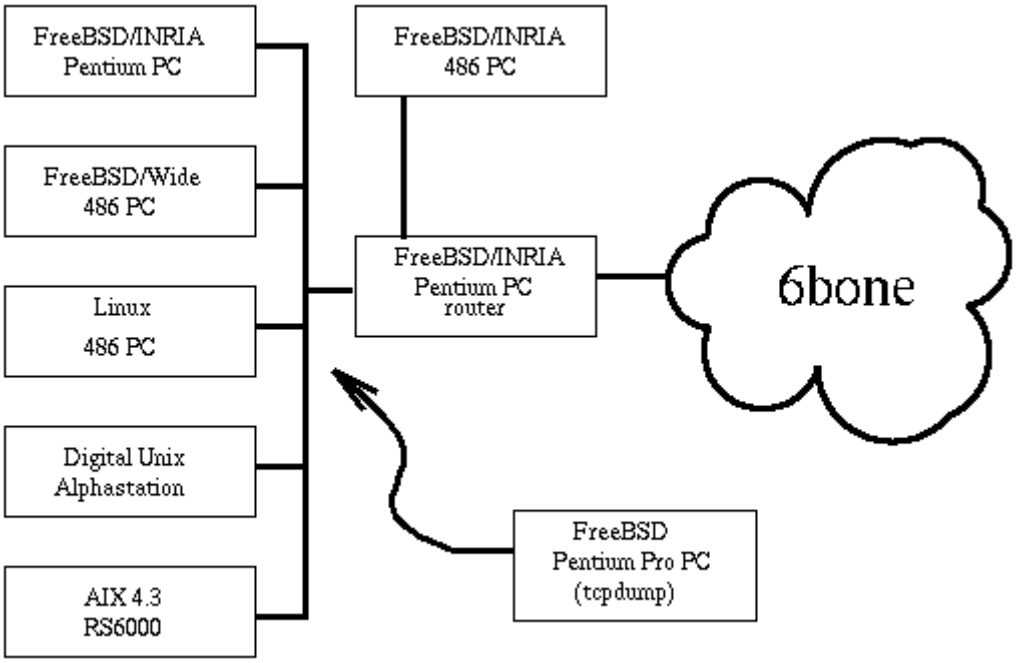

Figure 1.: Topology of the IPv6 test laboratory

Stateless autoconfiguration [1]
: The router was configured to advertise automatic configuration. We checked that the hosts set their link-local addresses correctly, and used this address in the following steps of the autoconfiguration.

Prefix discovery [2]
: The router advertised a prefix to be used on the link. A correctly operating host set a EUI-64 format unicast address -generated from the prefix and the interface's MAC address- to its interface.

Default route discovery [2]
: The router was configured to advertise itself as a default route. A properly working host had to set a default route to the router, using the router's link-local address as a gateway.

Link parameter discovery [2]
: The router advertised an MTU for the link. In this case every host is required to use this MTU.

Deprecated addresses [2]
: We have configured the router to advertise a prefix with non-infinite lifetime. After disabling the advertisement of this prefix, hosts had to cease using addresses with this prefix when its lifetime expired.

Multiple prefixes [2]
: The router was configured to advertise more than one prefix. Correctly operating hosts had to configure and use multiple addresses generated from these prefixes.

Duplicate address detection [2]
: We attempted to configure an address, that is already used on the same link, to an interface. The hosts had to perform a duplicate address detection, and reject this address.

Unreachability detection [2]
: In the Router Advertisement the value of retrans timer was set to a finite value. After the expiration of this time a host had to send a Neighbour Solicitation message, when attempting to contact other hosts. Correct operation was check using tcpdump.

Configured tunnels (IPv6 over IPv4)
: We checked whether the implementation supports configured tunnels.

Automatic tunnels (IPv4 using compatibility addressing)
: We examined whether the implementation is able to access IPv4 only nodes using IPv4

compatible IPv6 addresses.

Applications and services
: While IPv6 support in applications is not closely related to an implementation's IPv6 support, we have examined those applications, that were shipped with the package. These include the usual network utilities (telnet, ftp, mail, etc.)

Installation and documentation
: We have examined the ease of installation and the supplied documentation.

9. Results

Stateless autoconfiguration
: All implementations, with the exception of Secure Client have passed this test. Secure Client has no support for EUI-64 addresses.

Default route
: All systems set a default route to the router. The DEC implementation used the router's unicast address as well as its link-local address as a gateway.

Link parameter discovery
: The only faultless implementation was the INRIA one. The WIDE and AIX set and used the proper MTU, but neither ifconfig, nor netstat indicated this. However, using tcpdump the correct operation was determined. Linux and Digital Unix did accept the MTU advertisement, but used a different - smaller - value. This is a violation of RFC 1970.

Deprecated addresses
: Every implementation capable of automatic prefix configuration handled the deprecation correctly.

Multiple prefixes
: Every system was able to use multiple prefixes. The WIDE implementation paniced several times, when using more then one addresses. The ifconfig command of the AIX and WIDE indicated only the first unicast address for the interface, however, netstat showed all of them.

Duplicate addresses
: With the exception of the WIDE package, none of the implementation allowed the use of duplicate addresses. With the WIDE system, we have experienced unstable operation. With duplicate link-local addresses, the autoconf6 command of the Inria implementation stuck into an endless loop, but the duplicate address was not configured.

Unreachability detection
: All of the implementations were correct.

Configured tunnels
: Every system, except Secure Client, supported configured tunnels.

Automatic tunnels
: Secure Client did not accept IPv6 addresses, thus automatic tunnels could not be tested. All other systems supported them.

Applications
: At the application level, the INRIA kit has far the best support. Virtually all of the Berkeley network utilities supports IPv6 or have their IPv6 equivalent. Several applications' IPv6 port is included in the distribution. (http server, web browser, X window, etc.) The other implementations contain basic IPv6 applications. The Secure Client has barely more than kernel level IPv6 support.

  The Linux ftp server and client had interoperability problems with other systems, since it only supported the Short Port command, while the others (except the INRIA, which supported both) the Long Port. The table below indicates the most important IPv6 applications.

  DNS support is only half-implemented in every system, since none of them supports DNS over IPv6. Even though that the name server and the resolver handles AAAA records, DNS may only

be used over IPv4.

Legend

+	Supported
-	Not supported or not working
*	See text
?	Not yet tested
1...5	Grade, 5 is best

	FreeBSD INRIA	FreeBSD WIDE	Linux	Digital Unix	AIX	FTP Software Win95	FreeBSD KAME	Microsoft IPv6
Autoconfiguraton	+	+	+	+	+	+	+	+
Prefix discovery	+	+	+	+	+	-	+	+
Default route	+	+	+	*	+	+	+	+
Link parameters	+	*	-	-	*	-	?	?
Deprecated addresses	+	+	+	+	+	-	?	?
Multiple prefix	+	*	+	+	*	-	?	?
Duplicate addresses	*	*	+	+	+	-	?	?
Unreachability detection	+	+	+	+	+	+	?	?
Automatic tunnels	+	+	+	+	+	-	?	?
Configured tunnels	+	+	+	+	+	-	?	?
telnet	+	+	-	+	+	-	?	-
ftp	+	*	*	*	*	-	?	-
sendmail (SMTP)	+	-	+	-	-	-	?	-
http server	+	-	-	-	-	-	?	-
http client	+	-	-	-	-	-	?	-
NFS	+	-	-	-	-	-	-	-
DNS	*	*	*	*	*	-	?	?
X window	+	-	-	-	-	-	?	?
IPv6 applications included in the package	5*	3	2	3	3	1*	4	2
Ease of installation	4	2	1	5	5	5	4	?
On-line documentation	4	1	1	4	3	1	3	1
Documentation	1	1	1	5	2	1	1	2

Table 2.: Test Results

10. Summary

The IPv6 protocol is far from being mature. Many key standards and applications are under development. The IPv6 implementations try to follow this fast changing environment. As a consequence, the quality of the implementations vary considerably.

In general, the core IPv6 protocol is correctly implemented, thus there are no significant interoperability problems at the IP level. The 6bone proves this situation.

There are far greater differences when viewing the upper level services and applications. Some systems (AIX, Digital Unix, INRIA) have broad IPv6 support, while others (WIDE, Linux) contain enough for experimenting. FTP Software's Secure Client has only limited IPv6 functionality.

Based on our tests, the AIX, Digital Unix and Inria implementation may be used in "production" environments. The other implementations have some serious shortcomings.

However, we have to stress, that these observations are true at the moment of writing, and there is constant development and improvement.

11. References

[1] S. Thompson, T. Nartin, IPv6 Stateless Address Autoconfiguration , RFC1971 , August 1996.

[2] T. Narten, E. Nordmark, W. Simpson, Neighbor Discovery for IP Version 6 (IPv6) , RFC1970 , August 1996.

[3] M. Crawford, A Method for the Tranmission of IPv6 Packets over Ethernet Networks , RFC1972 , August 1996.

[4] R. Gilligan, S. Thomson, J. Bound, W. Stevens, Basic Socket Interface Extensions for IPv6 , RFC2133, May 1997.

[5] S. Thomson, C. Huitema, DNS Extensions to support IP version 6 , RFC 1886, December 1995.

[6] R. Hinden, S. Deering IP Version 6 Addressing Architecture , Internet-Draft, Nov 1997.

[7] R. Hinden, S. Deering Internet Protocol, Version 6 (IPv6) Specification , Internet-Draft, Nov 1997.

[8] FreeBSD INRIA implementation source, ftp://ftp.inria.fr/network/ipv6/freebsd

[9] FreeBSD WIDE homepage, http://www.wide.ad.jp/wg/ipv6/

[10] LINUX IPv6 FAQ, http://www.terra.net/ipv6/

[11] AIX version 4.3 (IBM), http://www.austin.ibm.com/software/OS/aix43.html

[12] Digital UNIX IPv6 KIT information, http://www.digital.com/info/ipv6/host-implementation.html

[13] The First Hungarian IPv6 network, http://www.ipv6.fsz.bme.hu/

[14] JOIN Tests of IPv6 Implementations,
http://www.join.uni-muenster.de/JOIN/ipv6/texte-englisch/tests.html

[15] KAME Project, http://www.kame.net/

[16] Microsoft Research IPv6, http://research.microsoft.com/msripv6/

[17] FTP software Secure Client for Windows 95, http://www.ftp.com/product/secure_client/

Vitae

János Mohácsi received the M. Sc. degree in computer engineering at the Technical University of Budapest, Hungary in 1993. Since then he is a researcher and assistant lecturer at the Department of Control Engineering and Information Technology (TUB), with special interest in networking technologies. He has been working with IPv6 since 1996.

Szabolcs Szigeti received the M. Sc. degree in electrical engineering at the Technical University of Budapest, Hungary in 1994. Since 1998 he is researcher and assistant lecturer at the Department of Control Engineering and Information Technology (TUB). He has been working with IPv6 since 1996.

Tamás Máray received the M. Sc. degree in electrical engineering at the Technical University of Budapest, Hungary in 1985. He is working at the Department of Control Engineering and Information Technology (TUB). He leads and takes part in several Internet and Web projects in the National Information Infrastructure Development Program in Hungary.