INTERNET-DRAFT
October 21, 1999

T. Jinmei, Toshiba
A. Onoe, Sony

An Extension of Format for IPv6 Scoped Addresses

<draft-ietf-ipngwg-scopedaddr-format-00.txt>

Status of this Memo

Abstract

This document defines an extension of the format for IPv6 scoped addresses. In the format, a scope identifier is attached to a scoped address in order to supplement the ambiguity of the semantics of the address. Using the format with some library routines will make scope-aware applications simpler.

1. Introduction

There are several types of scoped addresses defined in the "IPv6 Addressing Architecture" [ADDRARCH]. Since uniqueness of a scoped address is guaranteed only within the according scope, the semantics for a scoped address is ambiguous on a scope boundary. For example, when a user specifies to send a packet from a node to a link-local address of another node, the user must specify the link of the destination as well, if the node is attached to more than one link.

This characteristic of scoped addresses may introduce additional cost to scope-aware applications; a scope-aware application may have to provide a way to specify an instance of a scope for each scoped address (e.g. a specific link for a link-local address) that the

application uses. Also, it is hard for a user to "cut and paste" a scoped address due to the ambiguity of its scope.

This document defines an extension of the format for scoped addresses in order to overcome this inconvenience. Using the extended format with some appropriate library routines will make scope-aware applications simpler.

1.1 Requirements

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, if and where they appear in this document, are to be interpreted as described in [KEYWORDS].

2. Proposal

The proposed format for scoped addresses is <scoped_address><delimiter><scope_id> where <scoped_address> is a literal IPv6 address, <scope_id> is a string to identify the scope of the address, and <delimiter> is a string to distinguish between <scoped_address> and <scope_id>.

It can't be assumed that a same identifier is common to all nodes in the according scope. Hence the proposed format MUST be used only within a node and MUST NOT be sent on a wire. This also means that any character or string can be used as <delimiter> unless it conflicts with other symbols used in literal IPv6 addresses (for instance, ":" would be confusing and should not be used as <delimiter>.) However, the notation of <delimiter> and <scope_id> should be common within a single node.

The proposed format should be applied to multicast addresses as well as to unicast addresses.

Here are examples. If the dash character (`-') is used as <delimiter> and numerical identifiers are used as <scope_id>, the following addresses

```
fe80::1234 (whose link identifier is 1)
fec0::5678 (whose site identifier is 2)
ff02::9abc (whose link identifier is 5)
ff08::def0 (whose organization identifier is 10)
```

would be represented as follows:

```
fe80::1234-1
fec0::5678-2
ff02::9abc-5
ff08::def0-10
```

As mentioned above, other strings could be used as <delimiter> and <scope_id>. The addresses in the example could be represented as follows:

```
Address                 <delimiter>  <scope_id>
fe80::1234,ether1       `,'          "ether1"
fec0::5678%foo.com      `%'          "foo.com"
```

```
ff02::9abc++A           "++"         "A"
ff08::def0=ORG=XY-net   "=ORG="      "XY-net"
```

3. Interaction with API

The proposed format would be useful with some library functions defined in the "Basic Socket API" [BASICAPI], that translate a nodename to an address, or vice versa.

For example, if getaddrinfo() parses a literal IPv6 address in the proposed format and fill an identifier according to <scopde_id> in the sin6_scope_id field of a sockaddr_in6 structure, then an application would be able to just call getaddrinfo() and would not have to care about scopes.

Also, if getnameinfo() returns IPv6 scoped addresses in the proposed format, a user or an application would be able to reuse the result by a simple "cut and paste" method.

Note that these extensions to the API ensure lower compatibility and do not affect any existing applications.

4. Issues

This document does not define a specific string for <delimiter>, since it would be used only within a node and hence be implementation dependent. However, it might be better to define a standard one in order to ensure portability on various implementations.

In this document, it is assumed that an identifier of a scope is not necessarily common in the scope. However, it would be useful if some common notation is introduced (e.g. an organization name for a site). In such a case, the proposed format could be commonly used to designate a single interface (or a set of interfaces for a multicast address) in a scope.

When the network configuration of a node changes, the change may affect <scope_id>. Suppose that the case where numerical identifiers are sequentially used as <scope_id>. When a network card is newly inserted in the node, some identifiers may have to be renumbered accordingly. This would be inconvenient, especially when addresses with the numerical identifiers are stored in non-volatile storage and reused after rebooting.

There is the preferred format for literal IPv6 addresses in URL's [URLFORMAT]. When the preferred format is used for an IPv6 scoped address, it might have to be combined with the proposed format defined in the document. For example,

```
http://[fec0::1234-bar.com]:80/index.html
```

should be used instead of

http://[fec0::1234]:80/index.html

(in this example, the dash character '-' is used as <delimiter>, and the string "bar.com" is used as <scope_id>.) The combination would introduce some restrictions to <delimiter> and <scope_id>. For instance, `]' should not appear in <delimiter> nor in <scope_id>.

5. Implementation Experiences

The WIDE KAME IPv6 stack implements the extension to the getaddrinfo() and the getnameinfo() functions described in Section 3 of this document. The source code is available as free software, bundled in the KAME IPv6 stack kit.

The current implementation supports the extension only for link-local (unicast and multicast) addresses. Also, the implementation assumes that there is one-to-one mapping between links and interfaces, and hence it uses interface names as <scope_id> for links. As <delimiter> it uses the atmark(@).

For instance, the implementation shows its routing table as follows:

```
Internet6:
Destination      Gateway                     Flags  Intface
default          fe80::fe32:93d1@ef0         UG      ef0
```

This means that the default router is fe80::fe32:93d1 on the link identified by interface "ef0". A user can "cut and paste" the result in order to telnet to the default router like this:

```
% telnet fe80::fe32:93d1@ef0
```

6. Security Considerations

The use of this approach to represent IPv6 scoped addresses does not introduce any known new security concerns, since the use is restricted within a single node.

7. Authors' Addresses

Tatuya JINMEI
Research and Development Center, Toshiba Corporation
1 Komukai Toshiba-cho, Kawasaki-shi
Kanagawa 210-8582, JAPAN
Tel: +81-44-549-2238
Fax: +81-44-520-1841
Email: jinmei@isl.rdc.toshiba.co.jp

Atsushi Onoe
Internet Systems Laboratory, IN Laboratories, Sony Corporation
6-7-35 Kitashinagawa, Shinagawa-ku, Tokyo 141-0001, JAPAN
Tel: +81-3-5448-4620

Fax: +81-3-5448-4622
Email: onoe@sm.sony.co.jp

8. References

[ADDRARCH] Hinden, R., Deering, S., "IP Version 6 Addressing Architecture", RFC 2373, July 1998.

[BASICAPI] Gilligan, R. E., Thomson, S., Bound, J., Stevens, W., "Basic Socket Interface Extensions for IPv6", RFC 2553, March 1999.

[KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[STD-PROC] Bradner, S., The Internet Standards Process -- Revision 3, RFC 2026, October 1996.

[URLFORMAT] Hinden, R., Carpenter, B., Masinter, L., "Preferred Format for Literal IPv6 Addresses in URL's", Internet Draft, September 1999, <draft-ietf-ipngwg-url-literal-04.txt>