

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 2015

H. Kitamura  
NEC Corporation  
S. Ata  
Osaka City University  
M. Murata  
Osaka University  
July 2, 2014

**Free from Using Zone Identifier for IPv6 Link-Local Address**  
**<[draft-kitamura-ipv6-zoneid-free-03.txt](#)>**

**Abstract**

This document describes "Zone-ID Free" functions that make end users free from using zone identifiers (Zone-ID) for IPv6 link-local addresses.

When users deal with IPv6 link-local addresses, it is thought that it is mandatory thing to specify accompanied Zone-IDs. For end users, however, it is troublesome and nuisance thing to do it. Because it is very hard for normal end users to find appropriate Zone-IDs for this purpose.

From another viewpoint, the usage of IPv6 link-local addresses accompanied with Zone-IDs is quite different from the traditional usage of global addresses. Therefore many problems related with Zone-ID are caused and new specifications are required to fix these problems.

This document explores and describes how "Zone-ID Free" functions work and how end users are released from using Zone-IDs when they deal with IPv6 link-local addresses.

The "Zone-ID Free" functions are upper compatible with the current usages of dealing with IPv6 link-local addresses and harmless to the existing communications.

In order to obtain appropriate Zone-ID information, a new technology "Zone-ID Learning" that issues multiple probes is introduced.

## Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on April 2013.

## Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.



## Table of Contents

<a href="#">1. Introduction</a>	<a href="#">3</a>
<a href="#">2. Problem Analysis and Goals</a>	<a href="#">4</a>
<a href="#">2.1. Analysis of the problems of using Zone-ID</a>	<a href="#">4</a>
<a href="#">2.2. Goals of "Zone-ID Free" function and what will be done</a>	<a href="#">7</a>
<a href="#">3. Reconsideration of Zone-ID related issues from the beginning</a>	<a href="#">7</a>
<a href="#">3.1. Why we need Zone-ID?</a>	<a href="#">8</a>
<a href="#">3.2. Which actual process requires Zone-ID information?</a>	<a href="#">8</a>
<a href="#">3.3. Where is Zone-ID specification?</a>	<a href="#">9</a>
<a href="#">3.4. How Zone-ID info. dealt with in the current implementations.</a>	<a href="#">8</a>
<a href="#">4. Design and Implementation of "Zone-ID Free" function.</a>	<a href="#">10</a>
<a href="#">4.1. Resolve "sin6_scope_id" value by NO probing methods</a>	<a href="#">10</a>
<a href="#">4.1.1. One Case:</a>	<a href="#">10</a>
<a href="#">4.1.2. Self Case:</a>	<a href="#">10</a>
<a href="#">4.1.3. Filled Case:</a>	<a href="#">11</a>
<a href="#">4.2. "Zone-ID Learning"</a>	
Resolve "sin6_scope_id" value by issuing multiple probes	<a href="#">11</a>
<a href="#">4.3. Target Socket functions and Implementation</a>	<a href="#">14</a>
<a href="#">5. Considerations on "Zone-ID Learning"</a>	<a href="#">15</a>
<a href="#">6. Security Considerations</a>	<a href="#">16</a>
<a href="#">7. IANA Considerations</a>	<a href="#">16</a>
<a href="#">Appendix A. Implementations</a>	<a href="#">16</a>

**[1. Introduction](#)**

This document describes "Zone-ID Free" functions that make end users free from using zone identifiers (Zone-ID) for IPv6 link-local addresses.

When users deal with IPv6 link-local addresses, it is thought that it is mandatory thing to specify accompanied Zone-IDs. For end users, however, it is troublesome and nuisance thing to do it. Because it is very hard for normal end users to find appropriate Zone-IDs for this purpose.

From another viewpoint, the usage of IPv6 link-local addresses accompanied with Zone-IDs is quite different from the traditional usage of global addresses. Therefore many problems related with Zone-ID are caused and new specifications (such as [[RFC6874](#)]) are required to fix these problems.

The problems of using Zone-IDs are analyzed in the following section.



This document explores and describes how "Zone-ID Free" functions work and how end users are released from using nuisance Zone-IDs when they deal with IPv6 link-local addresses.

The "Zone-ID Free" functions are upper compatible with the current usages of dealing with IPv6 link-local addresses and harmless to the existing communications.

Since end users' nodes are not routers, network connecting topology of end users' nodes is simple. Even if their nodes are equipped multiple interfaces, they use only one interface to connect to network. Therefore, in most cases Zone-ID is easily obtained without issuing any probe packets.

For the rare cases that end users uses multiple interfaces simultaneously to connect to network, a new technology "Zone-ID Learning" that issues probe packets is needed.

## **2. Problem Analysis and Goals**

### **2.1. Analysis of the problems of using Zone-ID**

In this section, problems of using Zone-ID are analyzed and described.

Problem 1:

Zone-ID information is hard to tell for normal end users.

Current Zone-ID specifications require end users to do hard works. For normal end users who do not have enough knowledge of their nodes' interfaces and network configuration, it is hard to know Zone-IDs (names of their nodes' interfaces). Furthermore, it is too hard to find and tell which actual Zone-IDs are appropriate for their link-local communications.

It is very nuisance for normal end users, and it will be almost impossible for them to specify appropriate Zone-IDs.



## Problem 2:

Zone-ID is node-local info. and cannot be shared with others.

Zone-ID is local and closed information to each node. Zone-ID information used on one certain node is meaningless information for the other nodes. In other words, Zone-ID information cannot be shared with outside nodes. Zone-ID cannot be used for advertisement type information (e.g., printer or file folder sharing, DNS etc.).

Command line operations that include Zone-ID arguments also become local on one node. They cannot be reused on the other nodes. It means that "copy and paste" type operations will not work effectively for these command line operations.

Fig. 1 shows this typical problematic situation. [Server Node S] tries providing some kind of services (e.g., Web or file share) to clients via its Link-Local Address. Both [Client Node C1] and [C2] try accessing the Server's services. The provided services are the same, but Client Nodes can NOT use the same arguments (use different ones as Fig. 1 shows) to specify the same destinations.

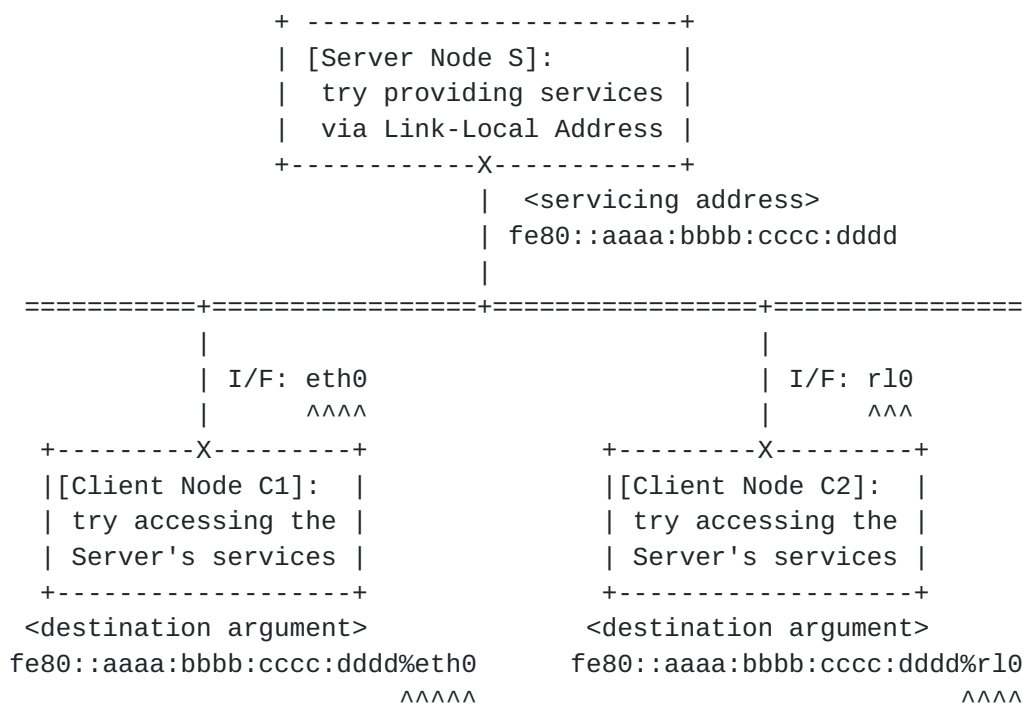


Fig. 1 Problem: on providing services via Link-Local Address





## Problem 3:

Zone-ID depends on node's situation and can be changed.

For an end user's node (e.g., Note PC) which has two (or more) interfaces (e.g., one is for Ethernet, the other is for Wireless LAN) and connects to the network via either of interfaces, used Zone-IDs for communications with the same destination link-local address are not always the same. It depends on the node's situation which interface is used to connect to the network. Even on the same node, this characteristic makes difficult to reuse command line operations that include Zone-ID arguments.

## Problem 4:

<Address>%<Zone-ID> representation is quite different from traditional <Address> only (without %<Zone-ID>) representation.

Since <IP address> is primitive information, it is used at various places everywhere of communication programs.

- Command line arguments of communication applications.
- "Address Bar" of Web browsers
- Configuration files
  - Firewall filter
  - DNS server
- in URI / URL
- etc.

The non-traditional (with %<Zone-ID>) representation affects to too many things. It is too hard and almost impossible to modify huge number of existing communication applications to accept <Address>%<Zone-ID> representation

Also, new specifications are required to handle <Address>%<Zone-ID> representation. [[RFC6874](#)] specification is designed only for URI. In order to cover all environments, more specifications are required.

Goals of this document are to make end users free from using Zone-ID and to provide environment that they do not require to accompany with %<Zone-ID> to their applications arguments.

(Non-Targets: people who have enough technical knowledge of networks and do not feel any difficulties to accompany with %<Zone-ID> information are not target. Also, a network environment that is too complicated to deal with for normal end users is not target.)



## **2.2. Goals of "Zone-ID Free" function and what will be done**

Fig. 2 shows current communication style. Every applications are required to accompany with "%<Zone-ID>" to their arguments.

```
-----  
> ping6 <Link-Local_Address_A>%<Zone-ID_X>  
> ping6 <Link-Local_Address_B>%<Zone-ID_Y>  
  
> ssh <Link-Local_Address_A>%<Zone-ID_X>  
> ssh <Link-Local_Address_B>%<Zone-ID_Y>  
-----
```

Fig. 2 Current Communication Style with %<Zone-ID>

Fig. 3 shows newly designed communication style that is achieved by "Zone-ID Free".

```
-----  
> ping6 <Link-Local_Address_A>  
> ping6 <Link-Local_Address_B>  
  
> ssh <Link-Local_Address_A>  
> ssh <Link-Local_Address_B>  
-----
```

Fig. 3 Newly Designed Communication Style  
without %<Zone-ID>

This is same to the traditional communication style (that is used in global (non link-local) address communications). This is simple and a kind of ideal communication style for normal end users.

Goals of the "Zone-ID Free" function are to achieve this simple and ideal communication style for IPv6 link-local addresses.

## **3. Reconsideration of Zone-ID related issues from the beginning**

As shown above, current specifications that require to end users to accompany with %<Zone-ID> imply many problems. We may have some prejudice on Zone-ID related specifications. In order to remove such prejudice, we have to reconsider Zone-ID related issues from the beginning.



### **3.1. Why we need Zone-ID?**

If a node has multiple interfaces, we cannot specify the destination correctly only with Link-Local Address information. Zone-ID (interface) information is also necessary to specify the destination uniquely.

This is true. We have to provide Zone-ID (interface) information to Link-Local Address communications.

Problems or difficulties of the current Zone-ID related specification should be located on the method that requires to !end users! to provide Zone-ID (interface) information.

We should find a method that provides Zone-ID (interface) information not only by end users but also by other parts.

### **3.2. Which actual process requires Zone-ID information?**

When receiving Link-Local packets, there are no difficulties. Zone-ID (interface) is easily and naturally identified from interface that is used by received packets.

When transmitting Link-Local packets, we meet a difficulty. Without Zone-ID (interface) information, we cannot transmit a packet.

Only when initially we transmit a Link-Local packet, we meet a difficulty.

All of IP communications are started after L2 addresses are resolved by Neighbor Discovery operations.

Therefore, initial transmitting operation can be concluded to transmitting NS (Neighbor Solicitation) packet operation and a difficulty is happened here.

In order to achieve "Zone-ID Free" function, it is necessary to improve current Neighbor Discovery operations. However, only with the ND improvement, "Zone-ID Free" function is not achieved.

In the following sections, additional necessary issues are discussed.



### **3.3. Where is Zone-ID specification?**

Basic Socket API [[RFC3493](#)] defines "sockaddr\_in6" structure as follows:

```
struct sockaddr_in6 {
    sa_family_t      sin6_family; /* AF_INET6 */
    in_port_t        sin6_port;   /* transport layer port # */
    uint32_t          sin6_flowinfo; /* IPv6 flow information */
    struct in6_addr    sin6_addr; /* IPv6 address */
    uint32_t          sin6_scope_id; /* set of interfaces for a scope */
};
```

"sin6\_scope\_id" member definition is directly related with Zone-ID information.

Socket functions that require "sockaddr\_in6" structure as their argument are related with Zone-ID specification.

(Advanced Socket API [[RFC3542](#)] also defines Zone-ID related specification. Targets of this document are simple usages of normal end users. Since Advanced Socket APIs are not used in the the simple usages, Zone-ID related specification that defined in [[RFC3542](#)], is not considered in this document.)

### **3.4. How Zone-ID info. dealt with in the current implementations.**

End user (application) must provide Zone-ID information to the kernel by filling "sin6\_scope\_id" member values.

When Zone-ID information is not provided by end user (application), "sin6\_scope\_id" member is filled with (special value) 0. When Socket functions are called, the kernel understands such a situation from value of "sin6\_scope\_id" member.

When the kernel receives "sin6\_scope\_id" = 0, the following two cases happen:

#### **1: Non Link-Local (such as Global) Address Case:**

Since Zone-ID information is NOT needed, Nothing happens.  
The kernel proceeds the operations.

#### **2: Link-Local Address Case:**

Since Zone-ID information is needed, problem happens.  
The kernel can not proceed the operations.  
Problem situation is reported to the calling Socket function.





Applications understand that problem happened at the kernel by the ERROR return value of the called Socket function, and applications cannot continue and STOP.

#### **4. Design and Implementation of "Zone-ID Free" function**

In order to achieve "Zone-ID Free" function, it is necessary to avoid happening the situation that Socket functions return ERROR value and applications STOP.

In other words, if an appropriate "sin6\_scope\_id" value is resolved at the kernel when the kernel receives "sin6\_scope\_id" = 0 in link-local address communications, the kernel can proceed the operations and applications will not stop. "Zone-ID Free" functions are achieved.

There are two types to resolve an appropriate "sin6\_scope\_id" value. One is NO probing (no probe needed) type. The other is probing (probe needed) type. In the following sections, they are discussed.

##### **4.1. Resolve "sin6\_scope\_id" value by NO probing methods**

There are three cases to resolve an appropriate "sin6\_scope\_id" value without issuing any probes. From a view point that no probes are needed, type of these cases can be called "trivial" type.

###### **4.1.1. One Case:**

If number of available interface of a node is One when Socket functions are called, there is no choice to select the one available interface for "sin6\_scope\_id" value.

This is very easy and there are no difficulties to implement this specification. Since it is popular that nodes have only one interface for end nodes, this case often happens and this specification works very effectively.

###### **4.1.2. Self Case:**

If the destination link-local address is issuer's Self address (set to interface of issuer's node), there is no choice to select the interface for "sin6\_scope\_id" value.

(Issued packets of Self case will be loopback ones.)



#### **4.1.3. Filled Case:**

If a Neighbor Cache entry of the destination link-local address has already been Filled with some values when Socket functions are called, we can use the filled interface information for "sin6\_scope\_id" value.

This situation means that the Neighbor Cache entry is filled by some operations before Socket functions are called.

It does not matter which operation is used to fill the entry, we just follow the filled information of the Neighbor Cache entry.

We mainly expect that the Neighbor Cache entry is filled by the multiple probes method which is described in the following [section 4.2](#).

#### **4.2. "Zone-ID Learning"**

##### **Resolve "sin6\_scope\_id" value by issuing multiple probes**

If the situation does not match any of above no probe needed cases (One/Self/Filled Cases), "sin6\_scope\_id (Zone-ID)" value must be resolved by probing.

At end user environment, Zone-ID information is usually obtained by either of One/Self/Filled Cases. So, it is very rare to execute "Zone-ID Learning" method.

As described in [Section 3.2](#), All of IP communications are started after L2 addresses are resolved by Neighbor Discovery operations. Therefore, initial transmitting operation can be concluded to transmitting NS (Neighbor Solicitation) packet operation.

With current NS packet transmitting operation, single packet is transmitted via the designated interface. And, replied NA packet that is received via the same interface is waited for.

If interface (Zone-ID) is not specified, there is no designated interface. With current operation, we could not transmit NS packet and problem happens.

In order to achieve "Zone-ID Free" function, we extend this operation. We will send multiple NS packets via available all interfaces. And replied NA packet that is received via either of interfaces that are used for transmitting the NS packets is waited for. After a NA packet is received, corresponding neighbor cache entry will be filled/updated.



Fig. 4 shows typical environment for "Zone-ID Learning".

[Initiator Node I] who has multiple(two) I/Fs (one faces Network X, the other faces Network Y) tries to send a packet to a Link-Local Address. Node does not know where the target LL address is located (on Network X or Y). So, the Node I send multiple NSs as probes on both Network X and Y, and wait for a replied NA from a Node who has the target LL address.

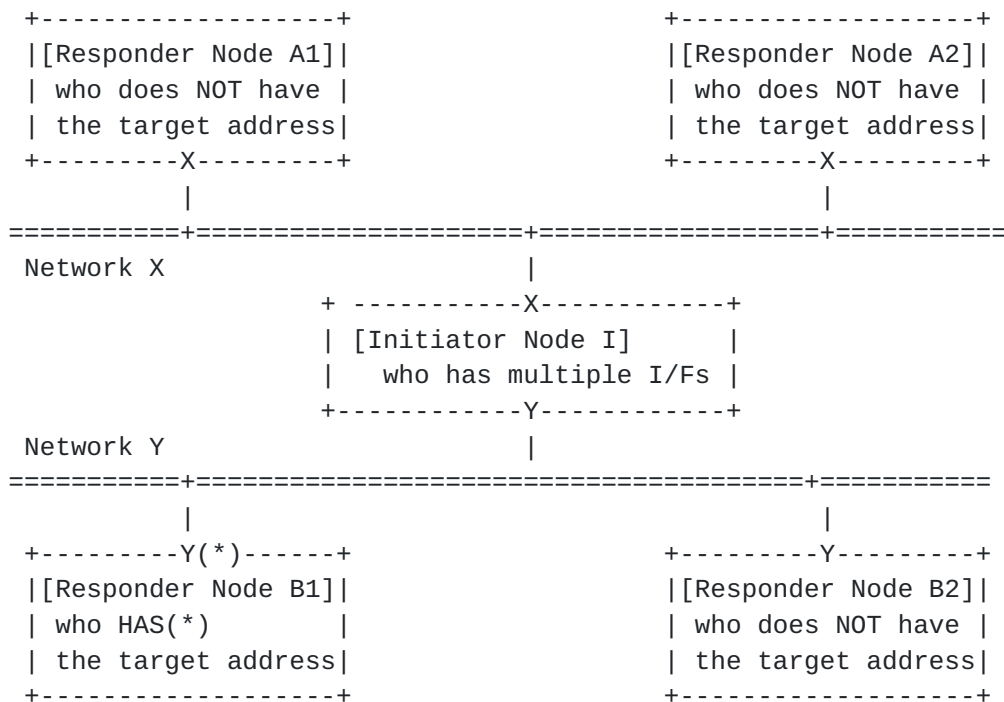


Fig. 4 "Zone-ID Learning": Typical Environment to send multiple problems

Typical sequence chart of the "Zone-ID Learning" mechanism is shown in the following Fig. 5.

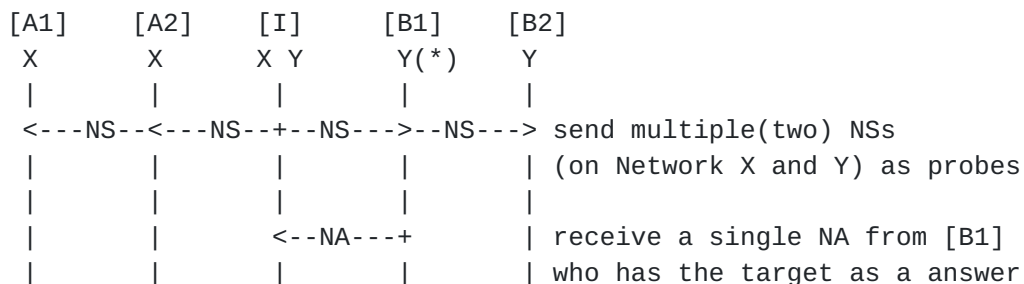


Fig. 5 "Zone-ID Learning": Typical Sequence Chart



Once neighbor cache entry is be filled, there will be no difficulty to issue IP packets.

Table 1 NS Transmit Specification Comparison

Specification	# of NS	via Interface (Zone-ID)
Current	Single	designated one interface
Zone-ID Learning	Multiple	available all interfaces

NS packet transmit specification comparison is shown at Table 1. By adding small improvement (multiple NS probing) to neighbor discovery implementation, this function is achieved.

Fig. 6 shows Command Line Operations enabled by above described "Zone-ID Free" functions.

```

-----
1: > ping6 <Link-Local_Address_C>

2: > ping6 <Link-Local_Address_C>
3: > ssh <Link-Local_Address_C>
4: > ... <Link-Local_Address_C>
-----

```

Fig. 6 Command Line Operations enabled by "Zone-ID Free"  
(First Probing and the followings Non-Probing)

Line 1(1:) and line 2(2:) are the same operations. Their arguments are the completely same. However, the internal kernel operates quite differently.

At the line 1(1:) operation, "sin6\_scope\_id" value is resolved by issuing multiple probes. At the following lines (2: 3: ...) operation, "sin6\_scope\_id" value is resolved by NO probing (Filled case).

Once neighbor cache entry is be filled by probing, probing procedures are not required in the the following operations.

If neighbor cache entry is vanished (by neighbor cache aging operations), operations are started over from the probing.





### **4.3. Target Socket functions and Implementation**

In order to achieve "Zone-ID Free" functions, socket functions that returns ERROR value when they are called with "sin6\_scope\_id" = 0 must be improved.

Therefore, the following 5 types of Socket functions that match above conditions are actual targets to be improved.

- 1: TCP connect()
- 2: UDP sendto()/sendmsg()
- 3: UDP connect()
- 4: TCP bind()
- 5: UDP bind()

Only with modifying the kernel implementation, socket functions do not returns ERROR value. So it is not necessary to modify userland/library implementation of Socket functions. Their corresponding kernel implementation of them must be modified.

- 1: TCP connect()
- 2: UDP sendto()/sendmsg()

Since goals of above 2 types of Socket functions is to issues IP packets, it is easy to associate that they can become trigger functions to issue multiple NS probing messages (when correspondent neighbor cache entry is empty). IP packets are issued soon after NS/NA sequences are finished.

- 3: UDP connect()

Since goal of UDP connect() functions is NOT to issues IP packets, it is not easy to associate that UDP connect() functions can become trigger functions to issue NS probing messages. However, in order not to return Error value and to resolve "sin6\_scope\_id" value, multiple NS probing messages can be issued here.

- 4: TCP bind()
- 5: UDP bind()

Since an argument of bind() function must be Self address, "sin6\_scope\_id" value is solved by no probing (Self Case).



## 5. Considerations on "Zone-ID Learning"

There are three cases to reply to multiple-probes that are issued by "Zone-ID Learning" method.

### No Reply:

If there is no node that has the destination Link-Local address on any neighbor links, this situation is happened. This case often happens. Since this is clear situation, there are no difficulties.

### Single Reply:

This is standard and majority case. If there is one node that has the destination Link-Local address on either neighbor links, this situation is happened.

This is expected most comfortable situation. Zone-ID information is resolved definitely. There is no are no difficulties, and this is frequently happened typical case.

### Multiple Replies:

This case is rarely happened. If a source node face to more than two neighbor links where a node that has the destination Link-Local address exist, this situation is happened.

Since this situation does not violate a rule that Link-local address is unique on each link, this is possible case.

If the link-local address value is non-manually set complicated one (such as EUI64-based or random), this case would not be happened. Because 64bit Interface-ID space is wide enough to avoid the address collision.

If the link-local address value is manually set human-rememberable simple one (such as fe80::1), this case may be happened.

If there are multiple NA replies when the "Zone-ID Learning" is executed, the corresponding multiple neighbor cache entries will be created. Which Zone-ID (Interface) is selected from them is depended on which neighbor cache search logic is implemented.

In this case, appropriate Zone-ID may not be selected (obtained). If appropriate Zone-ID is needed, users can fall back to current destination specification method that accompanies Zone-ID information. Since the "Zone-ID Free" functions are upper compatible with the current usages of dealing with IPv6 link-local addresses, there is no difficulties to do this.



This is the special case that the link-local address is set manually. So it will be endurable to fall back to the current inconvenient method at this case.

## 6. Security Considerations

Since the issuing multiple probes operation of "Zone-ID Free" function is implemented by extending Neighbor Discovery operation Security Considerations of Neighbor Discovery [[RFC4861](#)] can be also applied to this document.

## 7. IANA Considerations

This document does not require any resource assignments to IANA.

## Acknowledgment

A part of this work is supported by the program: SCOPE (Strategic Information and Communications R&D Promotion Programme) operated by Ministry of Internal Affairs and Communications of JAPAN.

## [Appendix A](#). Implementations

Currently, above described "Zone-ID Free" functions have been implemented and verified under the following OS.

Ubuntu 13.04 (kernel 3.8.13.8)

## References

### Normative References

- [RFC3513] R. Hinden and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", [RFC 3513](#), April 2003
- [RFC4007] S. Deering, B. Haberman, T. Jinmei, E. Nordmark and B. Zill, "IPv6 Scoped Address Architecture", [RFC 4007](#), February 2005.
- [RFC3493] R. Gilligan, S. Thomson, J. Bound, J. McCann and W. Stevens, "Basic Socket Interface Extensions for IPv6", [RFC3493](#), February 2003



[RFC4861] T. Narten, E. Nordmark, W. Simpson and H. Soliman,  
"Neighbor Discovery for IP Version 6 (IPv6)", [RFC 4861](#),  
September 2007

[RFC6874] B. Carpenter, S. Cheshire and R. Hinden, "Representing  
IPv6 Zone Identifiers in Address Literals and Uniform  
Resource Identifiers", [RFC 6874](#), February 2013

#### Informative References

[RFC3542] W. Stevens, M. Thomas, E. Nordmark and T. Jinmei,  
"Advanced Sockets Application Program Interface (API)  
for IPv6", [RFC 3542](#), May 2003

[RFC5952] S. Kawamura and M. Kawashima, "A Recommendation for IPv6  
Address Text Representation", [RFC 5952](#), August 2010.

#### Authors' Addresses

Hiroshi Kitamura  
Cloud System Research Laboratories, NEC Corporation  
(SC building 12F)1753, Shimonumabe, Nakahara-Ku, Kawasaki,  
Kanagawa 211-8666, JAPAN  
Phone: +81 44 431 7686  
Fax: +81 44 431 7680  
Email: kitamura@da.jp.nec.com

Shingo Ata  
Graduate School of Engineering, Osaka City University  
3-3-138, Sugimoto, Sumiyoshi-Ku, Osaka 558-8585, JAPAN  
Phone: +81 6 6605 2191  
Fax: +81 6 6605 2191  
Email: ata@info.eng.osaka-cu.ac.jp

Masayuki Murata  
Graduate School of Information Science and Technology, Osaka Univ.  
1-5 Yamadaoka, Suita, Osaka 565-0871, JAPAN  
Phone: +81 6 6879 4542  
Fax: +81 6 6879 4544  
Email: murata@ist.osaka-u.ac.jp



