Ad Hoc IP Address Autoconfiguration

draft-jeong-adhoc-ip-addr-autoconf-04.txt

Status of this Memo

By submitting this Internet-Draft, I certify that any applicable patent or other IPR claims of which I am aware have been disclosed, and any of which we become aware will be disclosed, in accordance with RFC3668.

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 August 14, 2005.

Copyright Notice

Copyright (C) The Internet Society (2005). All Rights Reserved.

Abstract
This document specifies the steps which a mobile node in ad hoc network takes in deciding how to autoconfigure its IPv4 or IPv6 address in network interface. Because the ad hoc IP address autoconfiguration in this document considers ad hoc network’s partition and mergence, the address duplication caused by ad hoc network’s mergence can be resolved through address resolution protocol. Also, this document specifies how to resolve the address duplication in order to guarantee the maintenance of upper-layer sessions, such as TCP session.

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [3].

Table of Contents

1. Introduction.......................................................3
2. Terminology........................................................3
3. Overview..........................................................4
4. Message Formats...................................................5
  4.1. Message Format for Ad Hoc IPv4 Address Autoconfiguration..5
  4.2. Message Format for Ad Hoc IPv6 Address Autoconfiguration..6
  4.3. Interface-Key Extension Format................................7
5. Ad Hoc IP Address Autoconfiguration..............................8
  5.1. Ad Hoc IPv4 Address Autoconfiguration..........................9
    5.1.1. Network Prefix for IPv4 Ad Hoc Network....................9
    5.1.2. Procedure of Ad Hoc IPv4 DAD..............................9
  5.2. Ad Hoc IPv6 Address Autoconfiguration........................11
    5.2.1. Network Prefix for IPv6 Ad Hoc Network...................11
    5.2.2. Procedure of Ad Hoc IPv6 DAD............................12
6. Maintenance of Upper-layer Session under Address Duplication...12
  6.1. Detection of Address Duplication during Weak DAD Phase....12
  6.2. Address Duplication Resolution................................13
  6.3. Data Packet Delivery after resolving Address Duplication.13
7. Open Issues..................................................................14
8. Configuration Parameters............................................14
9. Security Considerations.............................................14
10. Acknowledgements..................................................14
11. Normative References..............................................15
12. Informative References.............................................15
13. Authors’ Addresses................................................15
Intellectual Property Statement....................................16
Full Copyright Statement............................................17
Acknowledgement.....................................................17
1. Introduction

IPv6 stateless address autoconfiguration [4][5] provides a way to autoconfigure either fixed or mobile nodes with one or more IPv6 addresses and default routes. But this is not suitable for multi-hop ad hoc networks that have dynamic network topology. Ad hoc networks can become partitioned and merged frequently as intermediate nodes move. In this environment, IP address autoconfiguration should be able to deal with the address duplication not only within a connected ad hoc partition, but also in the case where two or more partitions having duplicate addresses respectively become merged. This document provides ad hoc IP address autoconfiguration in IPv4 ad hoc network as well as in IPv6 ad hoc network.

As we know from birthday paradox, there frequently happens an address conflict when each node chooses its address by random address selection in ad hoc network, especially in IPv4. In addition, due to network partitioning and merging, more address conflicts may occur. Therefore, the handling of address conflict, detection and resolution is very important in ad hoc IP address autoconfiguration based on random address selection. Because the ad hoc IP address autoconfiguration in this document considers ad hoc network’s partition and mergence, the address duplication that can be caused by ad hoc network’s mergence can be resolved through address resolution protocol. Also, this document specifies how to resolve the address duplication in order to guarantee the maintenance of upper-layer sessions, such as TCP session, with a minimum of packet loss.

2. Terminology

This document uses the terminology described in [4][5]. In addition, seven new terms are defined below:

Mobile Ad Hoc Network (MANET)

The network where mobile nodes can communicate with one another without preexisting communication infrastructure, such as base station or access point.

Duplicate Address Detection (DAD)

The process by which a node, which lacks an IP address, determines an address, determines whether the candidate address it has selected is available or not. A node already equipped with an IP address takes part in DAD in order to protect its IP address from being accidentally used by another node.

Strong DAD
The timed-based DAD for the purpose of checking if there is an address duplication in a connected MANET partition within a finite bounded time interval [6].

Weak DAD

The DAD for the purpose of detecting address duplication during ad hoc routing. A key is used for the purpose of detecting duplicate IP addresses, which is selected to be unique by mobile node. When mobile node receives a routing control packet, it compares the pairs of address and key contained in the packet with those in the routing table or cache [6].

Address Request (AREQ)

The message used during Strong DAD for the purpose of checking if there is another node having the requested address [7].

Address Reply (AREP)

The message used during Strong DAD for the purpose of indicating the requested address has already been utilized [7].

Address Error (AERR)

The message used during Weak DAD for the purpose of indicating that an address duplication happened or that the address of peer node has been changed.

3. Overview

IPv4 or IPv6 unicast address of ad hoc node can be autoconfigured by IP address autoconfiguration protocol for ad hoc networks. The configuration of address is comprised of three steps: (a) selection of a random address, (b) verification of the address uniqueness and (c) assignment of the address into network interface.

The duplication address detection (DAD) proposed in this document not only checks address duplication during the initialization of address configuration, but also checks and resolves address duplication detected by intermediate nodes during ad hoc routing. Also, even during the resolution of address conflict, the sessions using the conflicted address can still continue until the sessions are closed.

The DAD for ad hoc network in this document is a hybrid scheme consisting of two phases: (a) Strong DAD and (b) Weak DAD.
Within a connected ad hoc partition, Strong DAD can check quickly if there is any address duplication or not. During ad hoc routing, Weak DAD can find out if there is address duplication or not in the case where two or more MANET partitions having duplicate addresses are merged.

4. Message Formats

4.1. Message Format for Ad Hoc IPv4 Address Autoconfiguration

The mechanism of this document needs new ICMPv4 types for ad hoc IPv4 address autoconfiguration. Figure 1 shows the format of the messages related to IPv4 address autoconfiguration.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |      Code     |            Checksum           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Identification                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Originator’s IPv4 Address                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Requested or Duplicate IPv4 Address             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1. Message Format for Ad Hoc IPv4 Address Autoconfiguration

Fields:

- **Type**: 8-bit identifier of the type of ICMPv4 message.
  
<table>
<thead>
<tr>
<th>Message Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AREQ</td>
<td>(TBD)</td>
</tr>
<tr>
<td>AREP</td>
<td>(TBD)</td>
</tr>
<tr>
<td>AERR</td>
<td>(TBD)</td>
</tr>
</tbody>
</table>

- **Code**: 8-bit unsigned integer. As the code for message type, code values 1 and 2 in AERR message indicate the announcement and acknowledgement of address change, respectively. In the other cases, code value is always set to 0.

- **Checksum**: 16-bit unsigned integer. The checksum for the ICMPv4 message and parts of the IPv4 header

- **Identification**: 32-bit unsigned integer. The identification for ad hoc address autoconfiguration message is used
to prevent duplicate AREQ message from being rebroadcast.

Originator’s IPv4 Address
The IPv4 address of the sender of ad hoc address autoconfiguration message.

Requested or Duplicate IPv4 Address
The requested IPv4 address in AREQ and AREP messages, or the duplicate IPv4 address in AERR message.

AREQ and AREP messages are used during Strong DAD and AERR message during Weak DAD. Because AREQ message is forwarded by higher layer than network layer through local broadcasting, "Identification" field is necessary in order not to rebroadcast the message sent previously.

4.2. Message Format for Ad Hoc IPv6 Address Autoconfiguration

The mechanism of this document needs new ICMPv6 types for ad hoc IPv6 address autoconfiguration. Figure 2 shows the format of the messages related to IPv6 address autoconfiguration.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |      Code     |            Checksum           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Identification                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
+                    Originator’s IPv6 Address                  +
|                                                               |
| +                                                               |
| |                                                               |
| | +                                                               |
| |  |                                                               |
| |  +                                                               |
| |   | Requested or Duplicate IPv6 Address                      |
| |   +                                                               |
| |     +                                                               |
| +                                                               |

Figure 2. Message Format for Ad Hoc IPv6 Address Autoconfiguration
```
Fields:

- **Type**: 8-bit identifier of the type of ICMPv6 message.
  - **Message Name** | **Type**
    - AREQ (TBD)
    - AREP (TBD)
    - AERR (TBD)

- **Code**: 8-bit unsigned integer. As the code for message type, the valid value is 0, 1, or 2. Code value 1 in AERR message indicates that a node’s IP address has been changed. Code value 2 in AERR message indicates the acknowledgement for address change from the peer node. In the other cases, code value is always set to 0.

- **Checksum**: 16-bit unsigned integer. The checksum for the ICMPv6 message and parts of the IPv6 header.

- **Identification**: 32-bit unsigned integer. The identification for ad hoc address autoconfiguration message is used to prevent duplicate AREQ message from being rebroadcast.

- **Originator’s IPv6 Address**: The IPv6 address of the sender of ad hoc address autoconfiguration message.

- **Requested or Duplicate IPv6 Address**: The requested IPv6 address in AREQ and AREP messages, or the duplicate IPv6 address in AERR message.

### 4.3. Interface-Key Extension Format

A key for Weak DAD is contained in Interface-Key Extension of Figure 3, which is assigned to each network interface.

The Interface-Key extension is appended to control packets of ad hoc routing protocol for Weak DAD, such as route discovery message or hello message. Intermediate routing points MUST maintain the "Key" value for each address in routing table or cache.
5. Ad Hoc IP Address Autoconfiguration

The procedure of ad hoc IP address autoconfiguration in an ad hoc node is comprised of two phases: (a) Strong DAD phase and (b) Weak DAD phase. Especially, for Weak DAD, "Virtual IP Address" is used, which is the combination of "IP Address" and "Key". During ad hoc routing, with the value of Key, Weak DAD can detect IP address duplication. Therefore, Weak DAD places a requirement for a new field in the routing table -- namely, the inclusion of a "Key" field. Also, most of routing control packets of ad hoc routing protocols (e.g., link state packet) contain "Sequence Number" or "Identification" field in order to allow a receiving node of the control packets to determine whether it has recently seen copies of the packets. This field is also used for the purpose of detecting address duplication by Weak DAD.

Because this document does not consider the global connectivity to the Internet, it assumes that MANET is temporary network isolated from the Internet and the scope of addresses used in MANET is not global, but local.
5.1. Ad Hoc IPv4 Address Autoconfiguration

5.1.1. Network Prefix for IPv4 Ad Hoc Network

Among IPv4_MANET_PREFIX [7], IPv4 addresses in the range 1 ~ 2047 (TMP_ADDR) in the low-order 16 bits are used for temporary IPv4 unicast address during Strong DAD. The rest of addresses in the range TMP_ADDR + 1 ~ 65534 in the low-order 16 bits are used as tentative IPv4 address for actual IPv4 unicast address. After successful Strong DAD, the temporary address is replaced with the tentative address. In the future, this prefix can be replaced with another one for ad hoc network.

5.1.2. Procedure of Ad Hoc IPv4 DAD

During Strong DAD phase, an ad hoc node autoconfigures a unique IPv4 address in its network interface within a limited scope of a connected MANET partition and during Weak DAD phase, the node participates in Weak DAD which detects and deals with address duplication while ad hoc nodes exchange each other routing information, such as link state packet or route discovery packet, or broadcast their hello messages periodically.

The DAD procedure is as follows:

First of all, a node sets a variable for counting the number of Strong DAD’s failures, dad_count, to zero.

Step (a) : The node selects a temporary address and configures it in network interface.

Step (b) : The node selects a tentative address and makes an AREQ message for the address. It initializes a variable for retransmission of AREQ message, retrans_count, with zero. TTL of IP datagram for Strong DAD is set to TTL_STRONG_DAD. In proactive routing protocol, TTL of IP datagram MAY be set to one, one-hop distance.

Step (c) : The node broadcasts the AREQ message in IPV4_MANET_BROADCAST_ADDRESS and increases retrans_count by one. It waits for AREP message until the timer for Strong DAD expires. If an AREP message for the sent AREQ message arrives before the timer expires, the node executes Step (e). Otherwise, it executes Step (d). Notice that nodes under tentative state of Strong DAD for its address configuration SHOULD NOT participate in Strong DAD or routing.
Step (d) : If retrans_count is equal to DAD_RETRIES, indicating successful Strong DAD, the node goes to Step (f). Otherwise, it goes to Step (c).

Step (e) : If the AREP message received is associated with the AREQ message sent before and dad_count is unequal to DAD_FAILURE, the node increments dad_count by one and returns to Step (a) in order to restart Strong DAD for another address. Otherwise, the node reports error message and gives up its address autoconfiguration.

Step (f) : Because the requested address that is tentative is unique in the connected partition, the node replaces the temporary address with the tentatively selected address as a permanent IPv4 unicast address of its network interface.

Step (g) : The node waits for receiving address autoconfiguration messages or ad hoc routing control packets such as link state packet, route discovery packet and hello message. If the packet is address autoconfiguration message, it executes Step (h). If the received packet is ad hoc routing control packet, it executes Step (l).

Step (h) : First of all, it is checked during the processing of IP header of the message whether the message received is what was received previously on the basis of "Source IP Address" field of IP datagram containing the message and "Identification" field within the message or not. If the packet is what was received previously, the node discards the message, returning to Step (g). Otherwise, the node executes Step (i).

Step (i) : If the message is AREP, it executes Step (j). If the message is AERR, it executes Step (k). If the message is AREQ, the node compares the requested address in the AREQ message with its own address and active addresses in its routing table or cache. If an address duplication happens, it sends in unicast the originator node of the AREQ message an AREP message, indicating address duplication, returning to Step (g). Otherwise, it decrements the value of TTL of IP datagram, containing the AREQ message, by one and then rebroadcasts the message to neighbors, returning to Step (g).

Step (j) : If Destination IP address of the AREP message is the same as its own IP address and the duplicate address in the AREP message is corresponding to its own IP address under tentative state during Strong DAD, the node starts Strong DAD procedure again, that is returning to Step (a). For some reasons, if Destination IP address of IP header of the AREP message is the same as its own but the duplicate address in the AREP message is not corresponding to its own under tentative state during Strong DAD, it discards the message as error handling, returning to Step (g). Otherwise, it only relays the
message in unicast to next-hop node towards Destination IP address of
the AREP message, returning to Step (g).

Step (k) : If Destination IP address of the AERR message is the same
as its own IP address and the duplicate address in the AERR message
is the same as its own IP address, the node starts Strong DAD
procedure in order to autoconfigure a new address again, that is
returning to Step (a). In addition, in order to maintain the current
upper-layer sessions related to the duplicate address, it MAY inform
its peer nodes of address change. Refer to Section 6. For some
reasons, if Destination IP address of IP header of the AERR message
is the same as its own but the duplicate address in the AERR message
is not the same as its own, it discards the message as error handling
returning to Step (g). Otherwise, it only relays the message in
unicast to next-hop node towards Destination IP address of the AERR
message, returning to Step (g).

Step (l) : The node investigates each IP address contained in control
packet with Interface-Key extension to see whether for IP address,
there is a matching entry in routing table or cache. If there is a
matching entry and the values of Key associated with each address are
different, which means that an IP address conflict has happened, the
node sends in unicast an AERR message, indicating address conflict,
to another node using the duplicate address associated with less key
value, returning to Step (g). Otherwise, it executes the rest of the
procedure related to processing ad hoc routing control packets,
returning to Step (g).

Even in the accidental cases where the two contenders for an IP
address happen to select the same value for "Key", address
duplication MAY be detected with "Sequence Number" or
"Identification" field of the control packet. Assume that a node
receives a routing control packet (e.g., link state packet). If the
values of "IP Address" and "Key" fields within the packet are the
same as its own and the value of "Sequence Number" field within the
packet is higher than the counter value for its own "Sequence Number"
except sequence number wrap-around, the node MAY decide that address
duplication with the same key has happened and resolve the
duplication [8].

5.2. Ad Hoc IPv6 Address Autoconfiguration

5.2.1. Network Prefix for IPv6 Ad Hoc Network

Among the IPV6_MANET_PREFIX [7], "fec0:0:0:ffff::/96" is used as
IPV6_MANET_INIT_PREFIX for temporary unicast address during Strong
DAD. The low-order 32 bits of the temporary address are configured
with 32-bit pseudo random number. The rest of address range of
IPV6_MANET_PREFIX except IPV6_MANET_INIT_PREFIX is used for actual unicast address. The address is tentative address until the uniqueness of it is verified by Strong DAD. AREQ message for Strong DAD is broadcast in site-local scoped all node multicast address, IPV6_MANET_BROADCAST_ADDRESS.

Recently, IPv6 site-local address has been deprecated by IPv6 working group. Since IETF-56 meeting, IPv6 working group has been discussing local prefix for local networks separated from the Internet, such as ad hoc network [9]. If ad hoc prefix is determined by IPv6 working group, IPV6_MANET_PREFIX will have another for ad hoc network. IPV6_MANET_BROADCAST_ADDRESS will also be replaced with another for ad hoc network.

5.2.2. Procedure of Ad Hoc IPv6 DAD

An IPv6 ad hoc node autoconfigures a unique IPv6 address in its network interface in the same way as an IPv4 ad hoc node like section 5.1.2.

6. Maintenance of Upper-layer Session under Address Duplication

When address duplication happens and the duplicate address is replaced with another, the sessions above network layer, such as TCP session, can be broken. So, for the survivability of upper-layer sessions using the duplicate address, the notification of address change between the peer nodes is necessary. This resolution of duplicate address is more important than the detection of duplicate address from the viewpoint of network service; e.g., the maintenance of upper-layer sessions with a minimum of packet loss and delay.

6.1. Detection of Address Duplication during Weak DAD Phase

In order to allow data packets related to the sessions using the duplicate address to be forwarded to destination nodes for a while, after sending an error message (AERR) to the node related to the duplicate address, the intermediate nodes that have perceived address duplication should continue to forward on-the-fly data packets associated with the sessions using the duplicate address until the route entry for the duplicate address expires, only if there is one route entry towards the duplicate address. When there are more route entries than one associated with duplicate address of which keys are different each other, the intermediate nodes drop all the on-the-fly data packets so that the data packets may not reach a wrong destination and not perturb it.
Through this forwarding, the on-the-fly data packets of the node with duplicate address can be delivered to the destination without packet loss. For example, like in Figure 4, let’s assume that five nodes are connected to compose a MANET and node A is sending data packets to node E via node B, C and D. Even when the destination node E changes its address from X to Y due to address conflict, the on-the-fly data packets of the source node A can be delivered to node E without packet loss because the intermediate nodes can forward them because a route for node E’s duplicate address in each intermediate node is still valid.

6.2. Address Duplication Resolution

The node that receives an AERR message SHOULD autoconfigure a new IPv6 address through Strong DAD. Also, it SHOULD simultaneously allows the new address be used by the old upper-layer sessions using the duplicate address as well as by new upper-layer sessions from this time forward. The node SHOULD inform each peer node of its new address by sending an AERR message with code 1, which indicates its address change. The "Originator’s IPv4 Address" field of AERR message contains the duplicate address and the "Requested IPv4 Address" field contains a new address to be used for the further communication.

6.3. Data Packet Delivery after resolving Address Duplication

When the originator becomes to know that the sent AERR with code 1 has arrived at its peer node well after receiving an AERR with code 2 from the peer, it starts to send data packets to its peer node again with the new address through IP tunneling. The destination address in outer IP header is the new IP address of the node that announced duplicate address and that in inner IP header is the duplicate IP address of the node, i.e., the old address of the node. When the peer node receives tunneled packet from the sender, it decapsulates the packet and delivers the payload in the packet to upper-layer session associated with the duplicate address. Both the node and its peer node maintain the information of pairs of duplicate...
address and new address in Address Mapping Cache like a binding cache of Mobile IP [10][11] and use it for processing IP tunneling.

7. Open Issues

There might be some issues regarding Ad Hoc IP Address Autoconfiguration as follows:

- How to select victim node(s) under address conflict, considering the number of on-going sessions and fairness? The selection of victim node can affect network performance.

- How to implement data structure of the address mapping cache and how to maintain it?

8. Configuration Parameters

This section gives default values for some important parameters associated with Ad Hoc IP Address Autoconfiguration.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPV4_MANET_PREFIX</td>
<td>169.254/16</td>
</tr>
<tr>
<td>IPV6_MANET_PREFIX</td>
<td>fec0:0:0:ffff::/64</td>
</tr>
<tr>
<td>IPV6_MANET_INIT_PREFIX</td>
<td>fec0:0:0:ffff::/96</td>
</tr>
<tr>
<td>IPV4_MANET_BROADCAST_ADDRESS</td>
<td>255.255.255.255</td>
</tr>
<tr>
<td>IPV6_MANET_BROADCAST_ADDRESS</td>
<td>FF05::1</td>
</tr>
<tr>
<td>TTL_STRONG_DAD</td>
<td>3</td>
</tr>
<tr>
<td>DAD_RETRIES</td>
<td>3</td>
</tr>
<tr>
<td>DAD_FAILURE</td>
<td>3</td>
</tr>
</tbody>
</table>

9. Security Considerations

In order to provide secure ad hoc IP address autoconfiguration in ad hoc network, IPsec ESP MAY be used with a null-transform to authenticate ad hoc IP autoconfiguration messages or control packets, which can be easily accomplished through the configuration of a group pre-shared secret key for the trusted nodes.

10. Acknowledgements

The authors would like to acknowledge the previous contributions of the following people; Charles E. Perkins, Jari T. Malinen, Ryuji Wakikawa, Elizabeth M. Belding-Royer and Yuan Sun. In addition, the important definitions (e.g., Strong DAD and Weak DAD) and mechanisms for finding and resolving duplicate address have been derived from Nitin H. Vaidya’s work. Especially, we thank for his contribution.
For the suggestion of Passive DAD, in aid of Weak DAD, we thank Kilian Weniger.

11. Normative References


12. Informative References


13. Authors’ Addresses

Jaehoon Paul Jeong, Editor
ETRI/University of Minnesota at Twin Cities
117 Pleasant Street SE
Minneapolis, MN 55455
USA

Phone: +1 651 587 7774
EMail: jjeong@cs.umn.edu

Jungsoo Park
ETRI / PEC
161 Gajeong-dong, Yuseong-gu
Daegjeon 305-350
Korea

Phone: +82 42 860 6514
Fax: +82 42 861 5404
EMail: pjs@etri.re.kr

Hyoungjun Kim
ETRI / PEC
161 Gajeong-dong, Yuseong-gu
Daegjeon 305-350
Korea

Phone: +82 42 860 6576
Fax: +82 42 861 5404
EMail: khj@etri.re.kr

Hongjong Jeong
Kyungpook National University
1370 Sankyuk-dong, Puk-gu
Daegu 702-701
Korea

Phone: +82 53 940 8590
Fax: +82 53 957 4846
EMail: hjjeong@monet.knu.ac.kr

Dongkyun Kim
Kyungpook National University
1370 Sankyuk-dong, Puk-gu
Daegu 702-701
Korea

Phone: +82 53 950 7571
Fax: +82 53 957 4846
EMail: dongkyun@knu.ac.kr

Intellectual Property Statement

The following intellectual property notice is copied from RFC3668, Section 5.
The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Full Copyright Statement

The following copyright notice is copied from RFC3667, Section 5.4. It describes the applicable copyright for this document.

Copyright (C) The Internet Society (2005). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.