

MIF WG  
Internet-Draft  
Intended status: Informational  
Expires: January 2, 2015

H. Deng  
China Mobile  
S. Krishnan  
Ericsson  
T. Lemon  
Nominum  
M. Wasserman  
Painless Security, LLC  
July 1, 2014

Guide for application developers on session continuity by using MIF API  
draft-deng-mif-api-session-continuity-guide-04

## Abstract

Today most smart terminals are equipped with multiple interfaces such as 3G/LTE and WiFi, and users experience some loss of connectivity while switching interfaces. The MIF API draft [I-D.ietf-mif-api-extension] has specified an API to announce interface status information to the applications. Once the application receives such information, it can use this information to reconnect to its peer(s), and this could significantly improve the user experience.

## Requirements Language

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 [RFC2119].

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on January 2, 2015.

## Copyright Notice

Copyright (c) 2014 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. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Related MIF API information . . . . .	3
3. Using different source address to reconnect the server . . . . .	3
4. Generic guidelines for writing applications to handle new interfaces becoming available . . . . .	4
5. Generic guidelines for writing applications to handle interfaces becoming unavailable . . . . .	5
6. IANA Considerations . . . . .	5
7. Security Considerations . . . . .	5
8. Acknowledgements . . . . .	5
9. Normative References . . . . .	6
Authors' Addresses . . . . .	6

## 1. Introduction

A significant and increasing number of smart mobile terminals have multiple interfaces for connectivity (e.g. Wifi and 3G/LTE). These interfaces may have very characteristics in terms of reliability, available bandwidth, delay/jitter as well as cost per bit. There is some form of connection manager on the end device that picks an interface for communication based on some pre-configured policy and/or based on dynamic conditions. The initially selected interface may become deprioritized (e.g. due to a lower cost interface becoming available) or may become unavailable (e.g. due to loss of coverage when moving out of a WiFi hotspot). New interfaces may become available due to administrative action (e.g. manual activation of a specific connectivity technology) or due to dynamic conditions (e.g. entering coverage area of a wireless network or plugging in an ethernet cable). In order to handle such changes in connectivity, applications need to be aware of network status changes and react to them. This document provides a guide to writing such applications.

The MIF API [I-D.ietf-mif-api-extension] document specifies an API that is capable of providing information regarding changes in network and interface connectivity status. By using this information, application developers can develop applications that can survive changes in connectivity and even benefit from them.

The MIF MPVD Architecture [I-D.ietf-mif-mpvd-arch] document defines the notion of a PVD (set of network configuration information), and PVDs somehow must be exposed, in case applications are not PVD-aware, or indirectly participating the selection of PVD, or knowing of the PVDs based on PVD APIs. The MIF API [I-D.ietf-mif-api-extension] document specifies "Connect to PVD" message, application developers may develop application that can changes between different PVD connectivity.

## 2. Related MIF API information

MIF API draft [I-D.ietf-mif-api-extension] defines a few messages that are related to notifying whether an interface is available or not. The messages are defined in Section 3.5.1 (Announce Interfaces) and Section 3.5.4 (No Interface). Similar functionality is available for addresses using the messages defined in Section 3.5.12 (Announce Address) and Section 3.5.14 (No Address Announcement). The API also specifies interface change information in section 3.5.23.5 (Interface is going up) and 3.5.23.4 (Interface is going away). Both interface and address information could be used by the application to infer the availability of a new endpoint for communication or the loss of an existing endpoint for communication.

## 3. Using different source address to reconnect the server

The applications deployed on mobile hosts usually setup the connection with the server, then trying to keep the connection up as long as they can. This works reasonable well when the host has only one communication interface. Once the host has more than one communication interface, such as 3G/LTE and WLAN, such applications cease to work well. e.g. The per bit cost and the connection speed are different on these two interfaces, and the user would always prefer to change another cheaper and faster connection. e.g. While connecting to a WLAN interface after being connected to LTE, the mobile terminal would get a different set of configuration parameters including the IP address, DNS server and default gateway. Application would normally break after such change in connectivity if the original interface (3G/LTE) is turned off and manual intervention is usually required to reinitiate connectivity.

If the application is designed with changing network connectivity in mind, then the application could be carefully designed reconnect to

its peer based on MIF API notification about new interface(s) and/or new address(es). The application needs to start testing the usability of the new interface(s)/address(es) immediately and determine whether they are usable and, if so, decide what traffic to switch over. Please note that there are other solutions for handling address changes in the lower layers (network and transport) like MIPv6, shim6, and MPTCP that can shield the application from address changes. The guidelines provided in this document are also applicable when these techniques are being used. Also, there might be load balancers present on the server side and it may become very difficult to preserve sessions after an address change has occurred.

In most cases even when a mobile terminal gets WLAN connectivity and gets an IP address assigned, but it could still be disconnected from the Internet due to lack of authentication. As a consequence, the interface needs to be tested for internet connectivity before switching communication from an existing interface to a newly available interface.

4. Generic guidelines for writing applications to handle new interfaces becoming available

The recommended steps for the application developer to keep the session continuity based on MIF API are listed below:

Step 1: Application subscribes to the MIF API for interface and address change notifications;

Step 2: Application connects to the server based on interface 1 (either 3G/LTE or WLAN);

Step 3: When a new interface comes up or a new address is configured, the MIF API notifies the application.

Step 4: The application tries to re-connect to its peer from the newly available interface. If the connectivity check succeeds, then the application can successfully switch the communication over to the new interface based on policy or user initiated selection. Otherwise communication stays on the existing interface. The decision process on how a preferred interface is selected is out of scope of this document and might be the topic for a separate high level API document.

Step 5: The interface initially used for communication may now be turned off without disrupting communications if no other applications are using it.

## 5. Generic guidelines for writing applications to handle interfaces becoming unavailable

The recommended steps for the application developer to keep the session continuity based on MIF API are listed below:

Step 1: Application subscribes to the MIF API for interface and address change notifications;

Step 2: Application connects to the server based on interface 1 (either 3G/LTE or WLAN);

Step 3: When an interface or address, that is currently being used for communication, becomes unavailable the MIF API notifies the application.

Step 4: The application requests the MIF API to acquire a list of interfaces that are currently available. Based on locally configured preferences, the application tries to re-connect to its peer from one of the available interfaces. If the connectivity check succeeds, then the application can successfully switch the communication over to this interface.

Step 5: If the connectivity check fails, the application needs to redo the check for each of the available interfaces in order of preference until it can successfully connect to its peer.

Step 6: If at least one available interface is still able to connect to the peer, the application can switch over to this interface without disrupting communications.

## 6. IANA Considerations

This document does not require any IANA actions.

## 7. Security Considerations

Some applications may associate the the source address of the communication with the credentials used, it they may require refreshing the credentials after the application switches to using a new source address.

## 8. Acknowledgements

The authors would like to thank Pete McCann, Julien Laganier, Dapeng Liu, Dave Thaler, Brian Carpenter and Pierrick Seite for their comments and suggestions for improving this document.

## 9. Normative References

[I-D.ietf-mif-api-extension]

Liu, D., Lemon, T., Ismailov, Y., and Z. Cao, "MIF API consideration", draft-ietf-mif-api-extension-05 (work in progress), February 2014.

[I-D.ietf-mif-mpvd-arch]

Anipko, D., "MIF MPVD Architecture", draft-ietf-mif-mpvd-arch-01 (work in progress), May 2014.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

## Authors' Addresses

Hui Deng  
China Mobile  
No.32 Xuanwumen West Street  
Xicheng District,  
Beijing 100053  
China

Email: denghui02@gmail.com

Suresh Krishnan  
Ericsson  
8400 Blvd Decarie  
Town of Mount Royal, Quebec  
Canada

Email: suresh.krishnan@ericsson.com

Ted Lemon  
Nominum  
Redwood City,  
94063  
USA

Email: Ted.Lemon@nominum.com

Margaret Wasserman  
Painless Security, LLC  
356 Abbott Street,  
North Andover 01845  
USA

Email: [mrw@painless-security.com](mailto:mrw@painless-security.com)

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: Sep. 1, 2012

D. Liu  
China Mobile  
Ted. Lemon  
Nominum  
Yuri. Ismailov  
Ericsson  
Z. Cao  
China Mobile  
March 1, 2012

MIF API consideration  
draft-ietf-mif-api-extension-00

Abstract

This document describes an abstract API that provides the minimal functionality required for a program to communicate effectively with peers and services on the network while running on a host that has more than one active network interface. This API is abstract: we describe the functionality that must be provided, not the bindings that should be used to provide that functionality. The functionality described here provides the building blocks from which higher-level APIs might be built, and is not intended to be used directly by typical applications.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on May 3, 2012.

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. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	4
2. Conventions used in this document . . . . .	4
3. MIF API Concept . . . . .	5
3.1. Provisioning Domains . . . . .	5
3.2. Provisioning Domain Agnosticism . . . . .	5
3.3. MIF API Elements . . . . .	6
3.3.1. Application Element . . . . .	6
3.3.2. High Level API . . . . .	7
3.3.3. MIF API . . . . .	7
3.3.4. Communications API . . . . .	7
3.3.5. Network Link API . . . . .	7
3.4. MIF API communication model . . . . .	8
3.4.1. POST MESSAGE call . . . . .	8
3.4.2. CHECK MESSAGE call . . . . .	8
3.4.3. GET MESSAGE call . . . . .	8
3.5. MIF Messages . . . . .	8
3.5.1. Announce Interfaces . . . . .	9
3.5.2. Stop Announcing Interfaces . . . . .	9
3.5.3. Interface Announcement . . . . .	9
3.5.4. No Interface Announcement . . . . .	9
3.5.5. Announce Provisioning Domain . . . . .	9
3.5.6. Stop Announcing Provisioning Domains . . . . .	10
3.5.7. Provisioning Domain Announcement . . . . .	10
3.5.8. No Provisioning Domain Announcement . . . . .	10
3.5.9. Announce Configuration Element . . . . .	10
3.5.10. Configuration Element Announcement . . . . .	11
3.5.11. No Configuration Element Announcement . . . . .	11
3.5.12. Announce Address . . . . .	11
3.5.13. Address Announcement . . . . .	12
3.5.14. No Address Announcement . . . . .	12
3.5.15. Get Configuration Data . . . . .	12
3.5.16. Translate Name . . . . .	12
3.5.17. Stop Translating Name . . . . .	13
3.5.18. Name Translation . . . . .	13
3.5.19. Connect to Address . . . . .	13

3.5.20. Connect to Address From Address . . . . .	13
3.5.21. Connected . . . . .	14
3.5.22. Not Connected . . . . .	14
4. Example Usage . . . . .	14
5. Security Considerations . . . . .	16
6. IANA Considerations . . . . .	16
7. Acknowledgments . . . . .	16
8. References . . . . .	16
8.1. Normative References . . . . .	16
8.2. Informative References . . . . .	16
Authors' Addresses . . . . .	17

## 1. Introduction

Traditionally, hosts that communicate on the network have done so over a single network link, which is provided by a single service provider. This simple environment is relatively easy to program to, and relatively predictable.

However, this relatively simple case is no longer the norm. A typical modern host may have one or two wireless interfaces: a wireless interface connected to a broadband network, and possibly another connected to some kind of cellular network. The same host may also have a wired interface which is sometimes connected to another broadband link. It is also quite common for hosts to have VPN links that are configured, for example, for access to corporate networks, or for access to network privacy services.

As a result, it is now quite typical that a program attempting to communicate in such an environment will be presented with conflicting configuration information from more than one provider. In addition, the cost of bandwidth on different links and the power required by those links may require consideration.

The API specified in this document is intended to describe the minimal complete set of API calls required to implement higher level APIs that solve these problems. It is not expected that applications will be implemented to this API, although it should be possible to do so. Rather, we expect this API to be used as a basis for building higher-level APIs that provide domain-specific solutions to these problems. The reason for specifying a lower-level API is to enable any arbitrary domain-specific API to be implemented, since no single higher-level API is likely to satisfy the needs of every application.

The API specified here is an abstract API. This means that we specify the functionality that is required to implement the API, but we do not provide specific bindings for any programming language: these are left up to the implementation. The API is described in terms of messages sent and messages received, rather than in terms of procedure calls, because it is necessary to be able to interleave these messages; a procedure call API necessarily precludes interleaving.

## 2. 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 [RFC2119].

### 3. MIF API Concept

The MIF API is intended to deal with situations where more than one interface may be active at a time. It must also deal with situations where a single interface is connected to a link that provides more than one type of network service. The most common example of this that we expect is a dual-stack network configuration.

#### 3.1. Provisioning Domains

To properly handle these multiple-service interfaces, we specify the API not in terms of interfaces, but in terms of provisioning domains. So in the case of a dual-stack network attached to a single network interface, there would be two provisioning domains. If the host has a second interface that is connected to a link that only supports IPv6 service, then that host would be connected to a total of two network links, but three provisioning domains.

From the perspective of the MIF API, a provisioning domain consists of a link, plus all the configuration information received on that link for that provisioning domain. So for an IPv4 provisioning domain, that would be whatever information is received from the DHCP server. For an IPv6 provisioning domain, the information received through router advertisements would be combined with the information recieved via DHCPv6.

**\*\*point of discussion:** it's actually possible to have two separate provisioning domains for IPv6 on the same wire. Is this a case that could happen in practice, and that we ought to support? I know that some asian countries have arrangements where the operator of the physical network is distinct from one or more operators who provide transit; I think this is all handled transparently to the host, but I don't really know the details.

**\*\*point of discussion:** is IPv4 stateless/Bonjour a separate provisioning domain? What about IPv6 ULA?

#### 3.2. Provisioning Domain Agnosticism

Although it is possible that a high-level API built on top of this API may be able to distinguish between provisioning domains, at the level of this API, no such distinction can be made. Each provisioning domain is treated separately, and it is the responsibility of the higher-level API or of the application to decide which provisioning domain or domains to actually use.

### 3.3. MIF API Elements

There are a number of different, essentially independent, pieces of software that need to be connected together in order to fully support a successful MIF communication strategy. These elements are shown in figure 3.1.

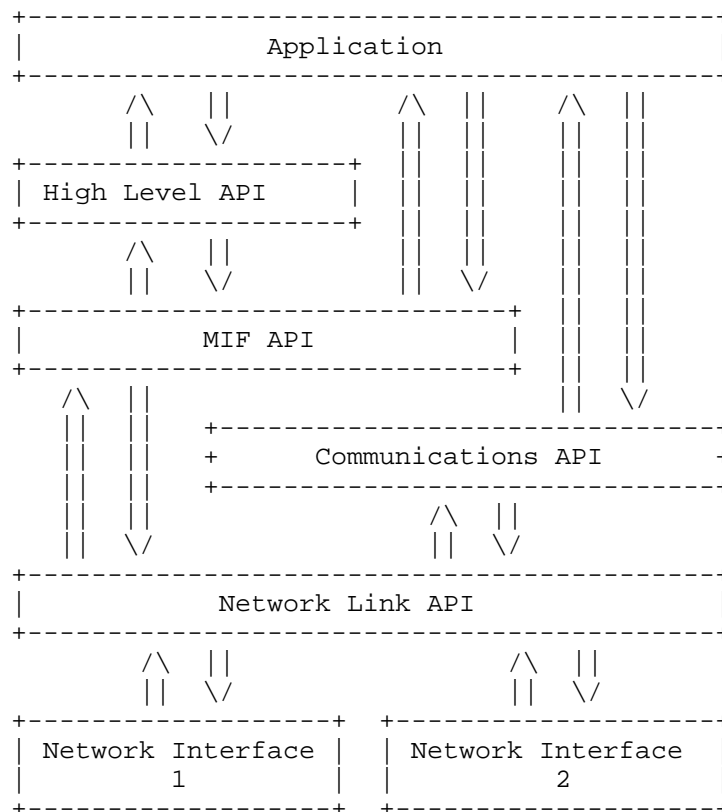


Figure 1

#### 3.3.1. Application Element

This is an actual application. Applications fall into a variety of broad categories, including network servers, web browsers, peer-to-peer programs, and so on. Although we are focusing here on the mechanisms required to allow these applications to originate connections to remote nodes, it is worth noting that applications must also be able to receive connections from remote nodes.

### 3.3.2. High Level API

Applications are generally expected to originate connections using some general-purpose high-level API suited to their particular function. It is likely that different applications may use different high-level APIs to communicate, depending on their particular needs. We do not describe the functioning of such high-level APIs; however, one such API under current consideration is the Happy Eyeballs for MIF [reference]. These APIs are expected to be able to be implemented using functionality like that described in the MIF API.

### 3.3.3. MIF API

This is the API being described in this document. Generally speaking, this API is used by higher-level APIs. However, it is permissible for applications to use the MIF API when it is deemed necessary. Currently, several modern web browsers take this approach to establishing network connections, rather than relying on vendor-provided connection mechanisms.

### 3.3.4. Communications API

Once an application has originated a connection with a remote node using either a high-level API or the MIF API, it must communicate. Similarly, when an application receives a connection from a remote node, it must communicate with that remote node. The communications API is used for this communication. Popular examples of such APIs include the POSIX socket API and a variety of other related APIs.

It is likely that in some instances, implementations of the MIF API will be done as extensions to the Communications API provided by a particular operating system; the functional separation we show here is intended to allow us to illustrate only those features required in a MIF environment, while relying on existing communications APIs to provide the rest.

### 3.3.5. Network Link API

This is the software that is responsible for actually managing whatever network links are present on a node, whether these are physical links or tunnels. What precisely this functional box contains may vary greatly from device to device. On a typical modern computer workstation, this functionality would almost certainly reside entirely in the system kernel; however, on an embedded device everything from the Application down to the Network Link API could easily be running together on the bare metal as a single program.

The Network Link API can completely concealed from the Application,

so we don't show a connection between them on the functional diagram, and indeed we do not talk about the functionality provided by this API. The reason for showing it on the functional diagram is simply to show that there likely is an API in common between MIF and the Communications API.

### 3.4. MIF API communication model

MIF API requests are made in the form of messages posted to the MIF API, and messages received from it. To accomplish this, several API calls are available. These calls mediate communication between the MIF API and the High Level API, or between the MIF API and the Application. In addition, the CHECK MESSAGE call allows the application to probe for or wait for messages from any of the APIs.

#### 3.4.1. POST MESSAGE call

This call causes a message to be posted to the MIF API. The call posts the message, and then returns.

#### 3.4.2. CHECK MESSAGE call

This call checks to see if there is a message waiting either from the High Level API, the MIF API, or the Communications API. Ideally it should be able to report the availability of any message or event that the application might anticipate receiving, so that the application can simply block waiting for such an event using this call. The application should be able to do a non-blocking probe, wait for some limited period of time, or wait indefinitely.

An example of a function of this type in existing practice is the POSIX poll() system call.

#### 3.4.3. GET MESSAGE call

This call checks to see if there is a message waiting. If there is no message, it returns a status code indicating that there is no message waiting. If there is a message, it returns the message.

### 3.5. MIF Messages

MIF messages always go in one direction or the other: from the subscriber to the MIF API, or to the subscriber from the MIF API. We use the term "subscriber" here to mean either the Application or the High Level API, since either is permitted to communicate with the MIF API.

Messages described here are grouped according to function.

### 3.5.1. Announce Interfaces

This message is sent to the MIF API to ask it to send a message announcing the existence of any interface. When the MIF API receives this message from a subscriber, it iterates across the list of all known interfaces; for each known interface, it sends an Interface Announcement message to the subscriber.

In addition, the MIF API sets a flag indicating that the subscriber is interested in learning about new interfaces. When the MIF API detects the presence of a new interface, it sends an Interface Announcement message for that interface to the subscriber. This would happen, for instance, when a new tunnel is configured, or when a USB device that is a network interface is discovered by the Network API.

Also, if a network interface goes away, either because the physical network device is disconnected, or because a tunnel is disabled, the MIF API will send a No Interface Announcement message to the subscriber.

### 3.5.2. Stop Announcing Interfaces

This message is sent to the MIF API when a subscriber is no longer interested in receiving announcements about new interfaces. Subsequently, the MIF API will no longer send Interface Announcement or No Interface Announcement messages to the subscriber.

### 3.5.3. Interface Announcement

This message announces the existence of an interface. The announcement includes an interface display name and interface identifier.

### 3.5.4. No Interface Announcement

This message announces that an interface that had been previously announced is no longer present. The announcement includes the interface identifier.

### 3.5.5. Announce Provisioning Domain

This message requests the MIF API to announce the availability of any provisioning domains configured on a particular interface. The interface identifier must be specified.

Upon receipt, the MIF API will iterate across the list of Provisioning Domains present for a particular interface, and will



send a Provisioning Domain Announcement for each such Provisioning Domain.

In addition, the MIF API will set a flag indicating that the subscriber wishes to know about new provisioning domains as they appear. Subsequently, when a new Provisioning Domain appears, the MIF API will send a Provisioning Domain Announcement message to the subscriber.

Finally, if a Provisioning Domain expires or is invalidated, the MIF API will send the subscriber a No Provisioning Domain Announcement message for that Provisioning Domain.

In the event that an interface on which provisioning domains has been announced goes away, a No Provisioning Domain Announcement message will be sent for each provisioning domain that had previously been announced on that interface before the No Interface Announcement message is sent.

Once a No Interface Announcement message has been sent, any subscriber that had subscribed to Provisioning Domain announcements for that interface will be automatically unsubscribed.

#### 3.5.6. Stop Announcing Provisioning Domains

This message requests that the MIF API stop sending the subscriber Provisioning Domain Announcement and No Provisioning Domain Announcement messages. The subscriber must indicate the interface for which it no longer wishes to receive Provisioning Domain announcements.

#### 3.5.7. Provisioning Domain Announcement

This message is sent by the MIF API to the subscriber to indicate that a new Provisioning Domain has successfully been configured on an interface. The announcement includes the interface identifier and the provisioning domain identifier.

#### 3.5.8. No Provisioning Domain Announcement

This message is sent by the MIF API to the subscriber to indicate that an existing, previously announced provisioning domain has expired or otherwise become invalid, and can no longer be used.

#### 3.5.9. Announce Configuration Element

This message is sent by the subscriber to request a specific configuration element from a specific provisioning domain. A

provisioning domain identifier must be specified.

The MIF API will respond by iterating across the complete list of configuration elements for a provisioning domain, sending a Configuration Element Announcement message to the subscriber for each one.

Additionally, if any Configuration Elements subsequently complete for a particular provisioning domain, the MIF API will send a Configuration Element Announcement message to the subscriber for each such element. If a Configuration Element becomes invalidated after it has been announced, the MIF API will send a No Configuration Element message.

If a provisioning domain expires or becomes invalid, the MIF API will iterate across the list of remaining configuration elements for that provisioning domain and send a No Configuration Element Announcement message for each such configuration element.

#### 3.5.10. Configuration Element Announcement

The Configuration Element Announcement message includes a Provisioning Domain ID and a Configuration Element Type, which can be one of the following:

- Config Element RA
- Config Element DHCPv6
- Config Element DHCPv4
- ...TBD...

#### 3.5.11. No Configuration Element Announcement

The No Configuration Element Announcement message indicates that a previously valid configuration element for a provisioning domain is no longer valid. The message includes a provisioning domain identifier and a configuration element type.

#### 3.5.12. Announce Address

This message is sent by the subscriber to request announcements of valid IP addresses for a specific provisioning domain. A provisioning domain identifier must be specified.

The MIF API will respond by iterating across the complete list of configuration elements for a provisioning domain, sending a Address Announcement message to the subscriber.

Additionally, if any new Address is subsequently configured on a particular provisioning domain, the MIF API will send an Address

Announcement message to the subscriber for each such element. If an address becomes invalidated after it has been announced, the MIF API will send a No Address Announcement message.

If a provisioning domain expires or becomes invalid, the MIF API will iterate across the list of remaining configuration elements for that provisioning domain and send a No Address Announcement message for each such address.

#### 3.5.13. Address Announcement

The Address Announcement message includes single IPv4 or IPV6 address and a Provisioning Domain identifier, as well as the valid and preferred lifetimes for that IP address (IPv6 only).

#### 3.5.14. No Address Announcement

The No Address Announcement message indicates that a previously valid address for a provisioning domain is no longer valid. The message includes a provisioning domain identifier and an IPv4 or IPv6 address.

#### 3.5.15. Get Configuration Data

The Get Configuration Data message is sent to the MIF API, and includes a Provisioning Domain ID, a Configuration Element Type, and a Configuration Information Identifier.

Configuration Information Identifiers:

- DNS Server List
- ...TBD...

The MIF API searches the configuration database for the specific type of Configuration Element on the specified Provisioning Domain to see if there is any configuration data of the specified type. If so, the MIF API sends a Configuration Data message to the subscriber; otherwise it sends a No Configuration Data message to the subscriber.

#### 3.5.16. Translate Name

The Translate Name message is sent to the MIF API. It includes a provisioning domain and a name, which is a UTF8 string naming a network node. The message also includes a Translation Identifier, which the subscriber must ensure is unique across all outstanding name service requests.

The MIF API begins a name resolution process. As results come in from the name resolution process, the MIF API sends Name Translation

messages to the subscriber for each such result.

Name resolution can be handled by one or more translations systems such as local host table lookup, Domain Name System, NIS, LLNMR, and is implementation-dependent. \*\*need to think about this

#### 3.5.17. Stop Translating Name

This message is sent to the MIF API to indicate that the subscriber is no longer interested in additional results from a particular name translation process. The message includes the Translation Identifier.

#### 3.5.18. Name Translation

The MIF API sends a Name Translation message to subscribers whenever results come in from a name translation process being performed on behalf of the subscriber. The Name Translation message includes the Translation ID generated by the subscriber, and an IP address returned by the translation process. If a single translation result contains more than one IP address, or IP addresses of different types, the MIF API sends a single Name Translation message for each such IP address.

#### 3.5.19. Connect to Address

The Connect to Address message contains an IP address, a provisioning domain identifier, and a connection identifier which the subscriber must ensure is unique. The MIF API attempts to initiate a TCP connection to the specified IP address using one or more source addresses that are valid for the specified provisioning domain, according to the source address selection policy for that provisioning domain.

If the connection subsequently succeeds, the MIF API will send a Connected message to the subscriber. If it subsequently fails, the MIF API will send a Not Connected message to the subscriber.

#### 3.5.20. Connect to Address From Address

The Connect to Address From Address message contains a source IP address, a destination IP address, a provisioning domain identifier, and a connection identifier which the subscriber must ensure is unique. The MIF API attempts to initiate a TCP connection to the specified IP address using the specified source address.

If the connection subsequently succeeds, the MIF API will send a Connected message to the subscriber. If it subsequently fails, the

MIF API will send a Connection Failed message to the subscriber.

#### 3.5.21. Connected

The Connected message contains the connection identifier that was provided in a previous Connect to Address or Connect to Address From Address message sent by the subscriber. It also contains an token, suitable for use with the connection API, for communicating with the end node to which the connection was established.

#### 3.5.22. Not Connected

The Not Connected message contains the connection identifier that was provided in a previous Connect to Address or Connect to Address From Address message sent by the subscriber. It also contains an indication as to what went wrong with the connection.

### 4. Example Usage

below is an example that shows how MIF API in use:

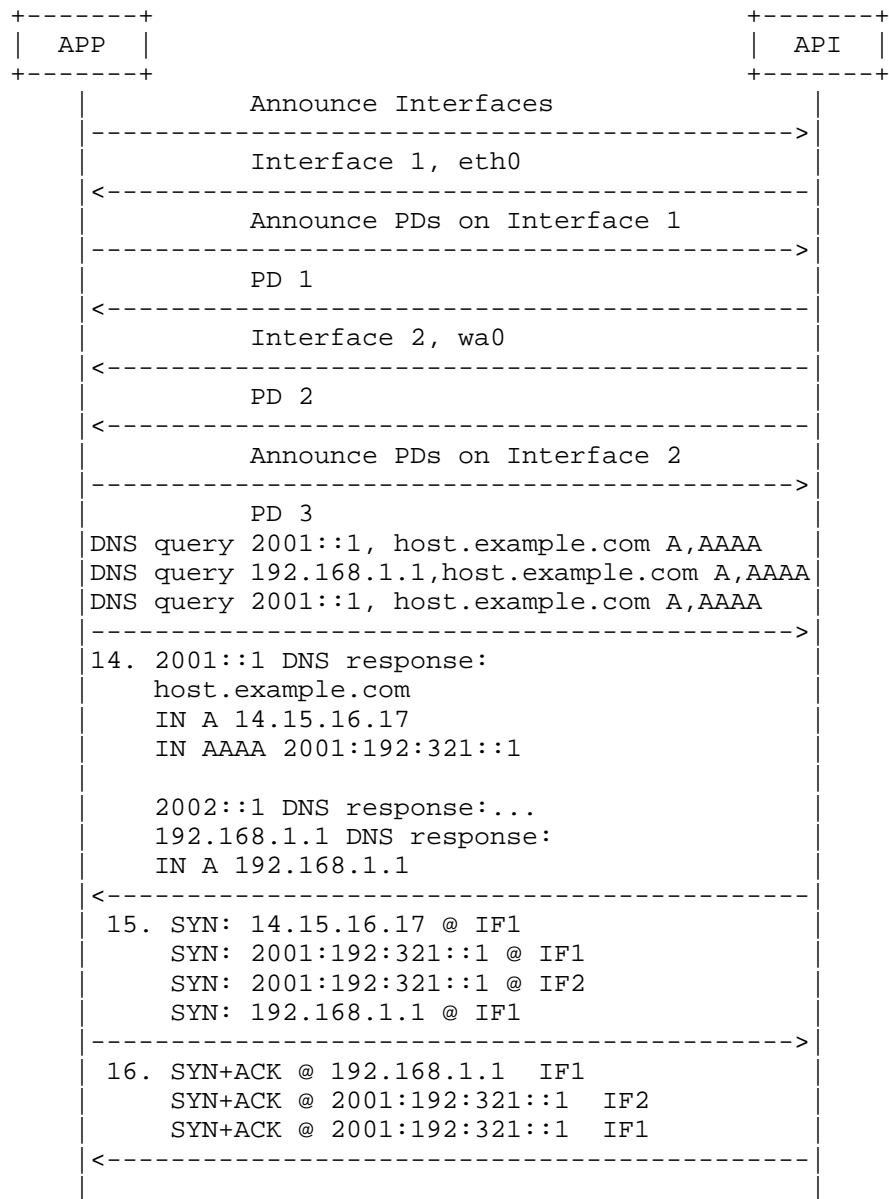


Figure 2

As described in the above communication model, the application first invoke the MIF API to query how many interfaces in the host. then, the application invokes MIF API to query how many networks attaches in each interface. application then invoke MIF API to query each DNS

configuration on each interface's attached network. application then send DNS query to each DNS server on each network. The DNS servers may return multiple IP address of the queried host name. The application then try to connect to each IP addresses of the host by sending tcp SYN packet to each destination IP addresses through multiple interfaces. Some of the destination IP address may return ACK packet some may not. The application then chose a best connection based on certain criteria. for example, the criteria may based on the quality of the link.

## 5. Security Considerations

TBD

## 6. IANA Considerations

None

## 7. Acknowledgments

The authors want to thank Teemu Savolainen from Nokia, Dayi Zhao from Bitway, Dave Thaler from Microsoft and others for their useful suggestions and discussions.

## 8. References

### 8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

### 8.2. Informative References

[I-D.scharf-mptcp-api]  
Scharf, M. and A. Ford, "MPTCP Application Interface Considerations", draft-scharf-mptcp-api-02 (work in progress), July 2010.

[RFC3493] Gilligan, R., Thomson, S., Bound, J., McCann, J., and W. Stevens, "Basic Socket Interface Extensions for IPv6", RFC 3493, February 2003.

Authors' Addresses

Dapeng Liu  
China Mobile  
Unit2, 28 Xuanwumenxi Ave,Xuanwu District  
Beijing 100053  
China

Email: liudapeng@chinamobile.com

Ted Lemon  
Nominum  
Redwood City  
CA 94063  
USA

Email: Ted.Lemon@nominum.com

Yuri Ismailov  
Ericsson  
Stockholm  
Sweden

Email: yuri@ismailov.eu

Zhen Cao  
China Mobile  
Unit2, 28 Xuanwumenxi Ave,Xuanwu District  
Beijing 100053  
China

Email: caozhen@chinamobile.com





Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: August 27, 2012

W. Dec  
Cisco Systems  
T. Mrugalski, Ed.  
ISC  
T. Sun  
China Mobile  
B. Sarikaya  
Huawei USA  
February 24, 2012

DHCPv6 Route Options  
draft-ietf-mif-dhcpv6-route-option-04

Abstract

This document describes DHCPv6 Route Options for provisioning IPv6 routes on DHCPv6 client nodes. This is expected to improve the ability of an operator to configure and influence a nodes' ability to pick an appropriate route to a destination when this node is multi-homed and where other means of route configuration may be impractical.

Requirements Language

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 [RFC2119].

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on August 27, 2012.

Copyright Notice

Copyright (c) 2012 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. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Problem overview . . . . .	3
3. Motivation . . . . .	4
3.1. Use cases . . . . .	4
3.2. Raised concerns . . . . .	9
3.2.1. Vendor-specific option . . . . .	9
3.2.2. Unicast RA . . . . .	9
3.2.3. DHCPv6 requires client to use one server . . . . .	10
3.2.4. Use VLANs . . . . .	10
4. DHCPv6 Based Solution . . . . .	11
4.1. Default route configuration . . . . .	11
4.2. Configuring on-link routes . . . . .	11
4.3. Deleting obsolete route . . . . .	11
4.4. Applicability to routers . . . . .	12
4.5. Updating Routing Information . . . . .	12
4.6. Limitations . . . . .	13
5. DHCPv6 Route Options . . . . .	13
5.1. Next Hop Option Format . . . . .	14
5.2. Route Prefix Option Format . . . . .	15
6. DHCPv6 Server Behavior . . . . .	16
7. DHCPv6 Client Behavior . . . . .	17
7.1. Conflict resolution . . . . .	18
8. IANA Considerations . . . . .	18
9. Security Considerations . . . . .	19
10. Contributors and Acknowledgements . . . . .	19
11. References . . . . .	20
11.1. Normative References . . . . .	20
11.2. Informative References . . . . .	20
Authors' Addresses . . . . .	21

## 1. Introduction

The Neighbor Discovery (ND) protocol [RFC4861] provides a mechanism for hosts to discover one or more default routers on a directly connected network segment. Extensions to the Router Advertisement (RA) protocol defined in [RFC4191] allow hosts to discover the preferences for multiple default routers on a given link, as well as any specific routes advertised by these routers. This provides network administrators with a new set of tools handle multi-homed host topologies and influence the route selection by the host. This ND based mechanism however is sub optimal or impractical in some multi-homing scenarios, where DHCPv6 [RFC3315] is seen to be more viable.

This draft defines the DHCPv6 Route Options for provisioning IPv6 routes on DHCPv6 clients. The proposed option is primarily envisaged for use by DHCPv6 client nodes that are capable of making basic IP routing decisions and maintaining an IPv6 routing table, broadly in line with the capabilities of a generic host as described in [RFC4191].

Throughout the document the words node and client are used as a reference to the device with such routing capabilities, hosting the DHCPv6 client software. The route information is taken to be equivalent to static routing, and limited in the number of required routes to a handful.

## 2. Problem overview

The solution described in this document applies to multi-homed scenarios including ones where the client is simultaneously connected to multiple access network (e.g. WiFi and 3G). The following scenario is used to illustrate the problem as found in typical multi-homed residential access networks. It is duly noted that the problem is not specific to IPv6, occurring also with IPv4, where it is today solved by means of DHCPv4 classless route information option [RFC3442], or alternative configuration mechanisms.

In multi-homed networks, a given user's node may be connected to more than one gateway. Such connectivity may be realized by means of dedicated physical or logical links that may also be shared with other users nodes. In such multi-homed networks it is quite common for the network operator to offer the delivery of a particular type of IP service via a particular gateway, where the service can be characterised by means of specific destination IP network prefixes. Thus, from an IP routing perspective in order for the user node to select the appropriate gateway for a given destination IP prefix,

recourse needs to be made to classic longest destination match IP routing, with the node acquiring such prefixes into its routing table. This is typically the remit of dynamic Internal Gateway Protocols (IGPs), which however are rarely used by operators in residential access networks. This is primarily due to operational costs and a desire to contain the complexity of user nodes and IP Edge devices to a minimum. While, IP Route configuration may be achieved using the ICMPv6 extensions defined in [RFC4191], this mechanism does not lend itself to other operational constraints such as the desire to control the route information on a per node basis, the ability to determine whether a given node is actually capable of receiving/processing such route information. A preferred mechanism, and one that additionally also lends itself to centralized management independent of the management of the gateways, is that of using the DHCP protocol for conveying route information to the nodes.

### 3. Motivation

The following section enumerates use cases, both in existing networks and as well as in envisaged future deployments. Usage scenarios are specified here in no particular order. As those use cases are described by various network operators, their scenarios may partially overlap.

Discussion: this section is rather long. Nevertheless, there were concerns raised that such option is not needed. Such extensive list can possibly solve those concerns. Number of use cases should be limited in future revisions. Alternatively, they can be moved to a separate motivation draft, if needed.

#### 3.1. Use cases

Use case 1: In Broadband network environment where the CPE is multi-homed to two upstream edge routers and each router provides connectivity for different types of services for example internet access and Video on Demand (restricted inside a walled garden) and the Service Provider would like to avoid routing on the CPE, there is a need to provision static route entries on RGs/CPEs. Service Provider requires a centralized control/management point for storing the customer's related information (IPv6 prefix, IPv6 routes and other provisioned information) and DHCPv6 is a good place for that. Using RA's would require to manually provision the edge router and this operation is not always possible, for example when router is operated by 3rd party. Broadband Forum document WT-124 issue 3 [BBF-WT-124] calls for this draft to solve the problem.

Use case 2: Operators want (approximate) feature parity so that they

can have (approximate) alignment between their operational procedures for v4 and v6, especially in a dual stack network. Having similar mechanisms for both protocols is desired due to lower operational expenses (OPEX).

Use case 3: In cellular networks, it is efficient for the network to configure routing information in central DHCPv6 server to do unified routing policy information. The gateways (GGSN in cellular network) only need to perform DHCPv6 relay. The Option code sent by clients can be used as an indication that host is MIF capable, so that network need not to do such configuration to host without MIF capabilities.

Use case 4: In cellular network, DHCPv6 is used for IPv6 parameter configuration and RA is used for SLACC of handset. This behavior was introduced in 3GPP Release 8 (or earlier). The network gateway in cellular network (e.g., GGSN) can naturally support DHCPv6 extension since the gateway acts as a DHCPv6 relay. However, it is very hard to update those gateways to use RA announcing the route information. The handsets with MIF feature need to visit subscribed/operator provided service. Some traffic is routed to the operator's network through 3G interface instead of to Internet through WiFi. DHCPv6 will be used to configure these specific routes. This use case is described in [THREEGPP-23.853].

Use case 5: PMIPv6 use case in LTE network. In LTE cellular network, both GTP and PMIPv6 are used for mobility management. In GTP, it is a point-to-point link between mobile host and PGW (PDN Gateway). However, in PMIPv6 case, the point-to-point link is between mobile host and SGW(Serving Gateway). The PGW sends /64 prefix to SGW through PBA. The SGW sends RA to mobile host. Route option may be needed when the host is multi-homed if it is simultaneously connected to the cellular network and WiFi or it simultaneously connects to multiple APNs in the cellular network. If RA is used for route configuration, both PGW and SGW(whose number is larger than PGW) need to be updated. Moreover, since a host can only connect to one SGW at a time, the SGW have to keep multiple route information received from different PGWs for one host and send them by RA to the host separately. This makes RA is not favorable in this use case.

Use case 6: WiFi networks. Some WiFi hotspots provide local services ("walled garden"). The route configuration on hosts or RGs is needed to direct some traffic to local network, while other traffic to the Internet. While this can be achieved using Route Information Option (RIO) in RA for all nodes that support [RFC4191], it does not allow doing so on a per-host basis.

Use case 7: VPN network. When a user connects to enterprise VPN

network, the routing of VPN traffic need to be configured. Due to the large number of such VPN networks, we cannot assume all the VPN network only use RA. DHCPv6 provides another choice which may be preferred by the VPN network. This situation is described in [RFC4191], Section 5.2. Hosts that do not support RFC4191 will not operate properly.

Use case 8: Selective walled garden. Figure 1 illustrates the case of two clients connected to a shared link. Both clients are assumed to have global IPv6 addresses and obtain their Internet connectivity via Router2 by means of a configured or a discovered default route. Client 1 however, unlike Client 2, is intended to run a specific application, e.g. VoIP, that is meant to access ServerA by means of Router1 with Server A being otherwise not reachable from the Internet. In addition to the global IP address Client1 may be assigned with another IP address of a more restricted scope for the purpose of communicating with Server A.

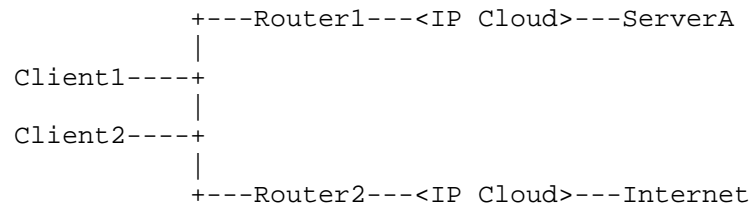


Figure 1: Walled garden scenario

The problem in the above scenario comes down to the fact that in order to reach Server A, Client1 requires to use a more specific route whose next-hop address is Router1. An ICMPv6 based mechanism for disseminating more specific route information, as defined in [RFC4191], disseminates this information via the shared link also to Client2. Often the operator wants to avoid this redundant dissemination to passing to Client2. In addition many operators prefer to be able to manage specific client route information from a centralized repository instead of managing directly such configuration on a router, as is required with the ICMPv6 based scheme. The former requirement is driven by the desire to provide to each client only the information required for their intended role which may be tied to a specific service, as well as to allow the possibility to introduce other routers into the scenario for purposes of load sharing. The requirement for more centralized configuration management is often due to administrative boundaries within an operator's organization as well as an existing operational practice that are in place for IPv4, all of which make router based configuration difficult.

Use case 9: Multihoming problem. A multihomed IPv6 host or gateway needs to solve at least 3 problems to operate properly when more than one link is operational:

1. Source address selection
2. Next-hop selection
3. DNS server selection

Problems one and three are solved by [I-D.ietf-6man-addr-select-opt] and [I-D.ietf-mif-dns-server-selection], respectively. It should be noted that both mechanisms use DHCPv6 as well. This draft attempts to solve problem two. Below is a brief explanation of the problem. See draft [I-D.ietf-v6ops-ipv6-multihoming-without-ipv6nat] for detailed problem analysis, background information and additional discussion regarding the need for a DHCPv6 solution to route information problem and IPv6 multihoming in general (with focus on aforementioned 3 problems).

In multihoming environment, server can restrict assignment of additional prefixes only to hosts that support more advanced next-hop and address selection requirements. (See Section 5.2 of [I-D.ietf-v6ops-ipv6-multihoming-without-ipv6nat]). Obviously this MUST be done on a per-host basis. Information about node capability is obtained via Option Request Option (ORO) in Solicit message, so support for Route Options is also used as means to report node capabilities to a network.

Use case 10: In static networks (i.e. networks that have static routers that are not changing over time, like home network with), such as some enterprise, hosting provider networks or even home network with a single router, it may be possible to stop using RA mechanism and deliver all configuration parameters to hosts using DHCPv6 only. This approach solves the rogue RA problem (i.e. a node that is not an approved router starts announcing RA in a network may hijack traffic from other hosts). This approach may be appealing in some cases, but not in all. For example if there is security association shared between clients and a DHCPv6 server, it may be useful to trust DHCP and disable RA mechanism. Also, environments that need DHCP for extended information, including but not limited to communicating information like DNS servers, hostnames, NTP servers, TFTP boot information and so on are forced to run two protocols increasing complexity and troubleshooting, where we have proof of concept in IPv4 that only one protocol (DHCP) should be needed.

Use case 11: It also has been proposed that route information option may be used as tie breaker in networks that deploy both DHCPv6 route



option and RA. DHCPv6 server could announce routing information along with RA. Legitimate router is also announced over DHCPv6. Host that receives conflicting information over RA may use additional information received from DHCPv6 as a tie breaker. This proposal [nanog-beijnum] was not investigated further.

Use case 12: DHCP-based configuration provides different failure mode than RA. While RA-based configuration works better in networks that offer redundant uplink using separate routers (second router can quickly take over upstream traffic), there are many deployments that cannot use that advantage, because of a single uplink. Current home networks with a single uplink as most obvious example. On the other hand, RA is more severely impacted by rogue entity problem. New rogue RA device may instantly break all other devices on the network. New rogue DHCP server will cause no immediate harm, may cause slow breakage over time, and may in fact never cause any breakage. This is due to the fundamental design choices of each protocol and it is hard to make either work the other way.

Use case 13: DHCP-based configuration may use mostly unicast traffic, while RA-based configuration mostly uses multicast. In some environments implementing multicast traffic may be cumbersome, e.g. in WiMAX environment not every subscriber station (SS) supports multicast channels and multicast capability must be emulated by base station (BS) using redundant transmissions. Classic, stateless, multicasted RA is in disadvantage compared to DHCP with standard unicast option enabled. While it is possible to selectively send unicasted RAs to selected subscribers, such architecture is essentially a stateful RA, thus forfeiting major benefit of RA being stateless.

Use case 14: Separated networks. In networks that do not have any routers, two DHCPv6 clients get a global address from DHCPv6 server. They cannot ping each other due to the fact that they do not know prefix that is available on-link. While it is tempting to suggest that separated networks should use link-local addressing, other factors should be taken into consideration. A stateful DHCPv6 may be used as a node monitoring tool, thus having advantage over link-local address usage. The also may be sensor networks that have outside connectivity only sporadically, e.g. uplink is established periodically to gather readings, but most of the time router is powered down for power reasons. Route Option in DHCPv6 could be used to configure on-link routes, while router could announce itself using short-lived RA.

Those requirements and use cases can be summarized as following:

1. In view of the DHCPv6 requirements in several fields, vendor-specific options lead to several segmented definitions. An IETF defined general option is a better choice.
2. Per user/host configuration makes DHCPv6 be used for the on-demand configuration.
3. As there is no well-defined central management system for prefix delegation and routing options via RA, it seems that DHCPv6 is the only available solution. It is better to have a generic option than a bunch of competing vendor options.
4. While this work was initially started with multihoming in mind, it is useful for single interface devices as well.

In a sense this route configuration mechanism makes DHCPv6 complete. Without it, this protocol cannot fully provision all configuration parameters to a host on its own.

### 3.2. Raised concerns

Opponents of this option proposed several alternative approaches. This section attempts to address raised issues.

#### 3.2.1. Vendor-specific option

Claim: During discussion about route configuration, some opponents say that routing information should be defined as vendor specific option.

Response: There are many ISPs, cellular and BBF network operators, CPE vendors, hardware vendors, DHCP implementors that want to implement and deploy this mechanism. Using vendor-specific option would severely limit interoperability and would make adoption and deployment much more complicated.

This solution is not a technology-specific requirement, it is requested by wide variety of companies, so it is not a vendor specific.

#### 3.2.2. Unicast RA

Claim: Some proponents insist that instead of using DHCPv6 solution, RA should be used instead. Some propose to send unicast RA with RIO option on a per-host basis.

Response: While this approach technically does not violate existing specs, it uses RA in a stateful way, thus the benefit of RA being

stateless is lost. Furthermore, it would require deploying additional mechanism, like RADIUS to deliver necessary information about hosts to routers. Authors consider deploying such stateful RA server with RADIUS support more complicated to deploy than the solution it tries to avoid (DHCPv6).

As there is no well-defined central management system for prefix delegation and routing options via RA, it seems that DHCPv6 is the only available solution. It is better to have a generic option than a bunch of competing vendor options.

Another concern raised is that RIO is not mandatory nor optional in 3GPP system and there is currently not support in 29.061 RADIUS or Diameter profile, so use of that alternative is somewhat limited in some cases.

### 3.2.3. DHCPv6 requires client to use one server

Claim: DHCPv6 has less rich semantics as client has to pick one out of all available server.

Response: While that is how currently most clients are implemented, there is nothing in [RFC3315] that mandates that. It is true that DHCPv6 was not designed with several provisioning domains. On the contrary, section 17.1.3 states that "Upon receipt of one or more valid Advertise messages, the client selects one or more Advertise messages based upon the following criteria.". This means that DHCPv6 client can obtain parameters from all available DHCPv6 servers, not just selected one. As such, DHCPv6 may work with overlapping provisioning domains. Authors acknowledge that this possibility is currently rather theoretical, as most known implementations do not take advantage of that possibility.

### 3.2.4. Use VLANs

Claim: There was a proposal to use VLANs as a solution to lack of per-host capability in RA mechanism.

Response: Deploying VLANs complicates network topology much more than adding a single DHCPv6 option. Furthermore in many cases it is not possible to deploy VLANs in any reasonable way, e.g. in multihost environment. Also, low cost devices (e.g. CPE) often do not offer VLAN capabilities, but they are very much capable of supporting DHCPv6. Another objection of esthetic nature. Using layer 2 mechanisms to work around limitations in layer 3 is not elegant.

#### 4. DHCPv6 Based Solution

A DHCPv6 based solution allows an operator an on demand and node specific means of configuring static routing information. Such a solution also fits into network environments where the operator prefers to manage Residential Gateway (RG) configuration information from a centralized DHCP server.

[I-D.ietf-v6ops-ipv6-multihoming-without-ipv6nat] provides additional background to the need for a DHCPv6 solution to the problem.

In terms of the high level operation of the solution defined in this draft, a DHCPv6 client interested in obtaining routing information request the route options using the DHCPv6 Option Request Option (ORO) sent to a server. A Server, when configured to do so, provides the requested route information as part of a nested options structure covering; the next-hop address; the destination prefix; the route metric; any additional options applicable to the destination or next-hop.

##### 4.1. Default route configuration

Defined mechanism may be used to configure default route. Default route is configured using RT\_PREFIX option that specifies ::/0 route, included as suboption in NEXT\_HOP.

Server MUST NOT define more than one default route.

##### 4.2. Configuring on-link routes

Server may also configure on-link routes, i.e. routes that are available directly over the link, not via routers. To specify on-link routes, server MAY include RTPREFIX option directly in Advertise and Reply messages.

##### 4.3. Deleting obsolete route

There are two mechanisms that allow removing a route. Each defined route has a route lifetime. If specific route is not refreshed and its timer reaches 0, client MUST remove corresponding entry from routing table.

In cases, where faster route removal is needed, server SHOULD return RT\_PREFIX option with route lifetime set to 0. Client that receives RT\_PREFIX with route lifetime set to 0 MUST remove specified route immediately, even if its previous lifetime did not expire yet.

#### 4.4. Applicability to routers

Contrary to Router Advertisement mechanism, defined in [RFC4861] that explicitly limits configuration to hosts, routing configuration over DHCPv6 defined in this document may be used by both hosts and routers. (This limitation of RA mechanism was partially lifted by W-1 requirement formulated in [RFC6204].)

One of the envisaged usages for this solution are residential gateways (RG) or Customer Premises Equipment (CPE). Those devices very often perform routing. It may be useful to configure routing on such devices over DHCPv6. One example of such use may be a class of premium users that are allowed to use dedicated router that is not available to regular users.

#### 4.5. Updating Routing Information

Network configuration occasionally changes, due to failure of existing hardware, migration to newer equipment or many other reasons. Therefore there a way to inform clients that routing information have changed is required.

There are several ways to inform clients about new routing information. Every client SHOULD periodically refresh its configuration, according to Information Refresh Time Option, so server may send updated information the next time client refreshes its information. New routes may be configured at that time. As every route has associated lifetime, client is required to remove its routes when this timer expires. This method is particularly useful, when migrating to new router is undergoing, but old router is still available.

Server MAY also announce routes via soon to be removed router with lifetimes set to 0. This will cause the client to remove its routes, despite the fact that previously received lifetime may not yet expire.

Aforementioned methods are useful, when there is no urgent need to update routing information. Bound by timer set by value of Information Refresh Time Option, clients may use outdated routing information until next scheduled renewal. Depending on configured value this delay may be not acceptable in some cases. In such scenarios, administrators are advised to use RECONFIGURE mechanism, defined in [RFC3315]. Server transmits RECONFIGURE message to each client, thus forcing it to immediately start renewal process.

See also Section 4.6 about limitations regarding dynamic routing.

#### 4.6. Limitations

Defined mechanism is not intended to be used as a dynamic routing protocol. It should be noted that proposed mechanism cannot automatically detect routing changes. In networks that use dynamic routing and also employ this mechanism, clients may attempt using routes configured over DHCPv6 even though routers or specific routes ceased to be available. This may cause black hole routing problem. Therefore it is not recommended to use this mechanism in networks that use dynamic routing protocols. This mechanism SHOULD NOT be used in such networks, unless network operator can provide a way to update DHCP server information in case of router availability changes.

Discussion: It should be noted that DHCPv6 server is not able to monitor health of existing routers. As there are currently more than 60 options defined for DHCPv6, it is infeasible to implement mechanism that would monitor huge set of services and stop announcing its availability in case of service outage. Therefore in case of prolonged unavailability human intervention is required to change DHCPv6 server configuration. If that is considered a problem, network administrators should consider using other alternatives, like RA and ND mechanisms (see [RFC4861]).

User is also encouraged to read Section 3.2.

#### 5. DHCPv6 Route Options

A DHCPv6 client interested in obtaining routing information includes the NEXT\_HOP and RT\_PREFIX options as part of its Option Request Option (ORO) in messages directed to a server (as allowed by [RFC3315], i.e. Solicit, Request, Renew, Rebind or Information-request messages). A Server, when configured to do so, provides the requested route information using zero, one or more NEXT\_HOP options in messages sent in response (Advertise, and Reply). So as to allow the route options to be both extensible, as well as conveying detailed info for routes, use is made of a nested options structure. Server sends one or more NEXT\_HOP options that specify the IPv6 next hop addresses. Each NEXT\_HOP option conveys in turn zero, one or more RT\_PREFIX options that represents the IPv6 destination prefixes reachable via the given next hop. Server includes RT\_PREFIX directly in message to indicate that given prefix is available directly on-link. Server MAY send a single NEXT\_HOP without any RT\_PREFIX suboptions or with RT\_PREFIX that contains ::/0 to indicate available default route. The Formats of the NEXT\_HOP and RT\_PREFIX options are defined in the following sub-sections.

The DHCPv6 Route Options format borrows from the principles of the Route Information Option defined in [RFC4191].

### 5.1. Next Hop Option Format

Each IPv6 route consists of an IPv6 next hop address, an IPv6 destination prefix (a.k.a. the destination subnet), and a host preference value for the route. Elements of such route (e.g. Next hops and prefixes associated with them) are conveyed in NEXT\_HOP option that contains RT\_PREFIX suboptions.

The Next Hop Option defines the IPv6 address of the next hop, usually corresponding to a specific next-hop router. For each next hop address there can be zero, one or more prefixes reachable via that next hop.

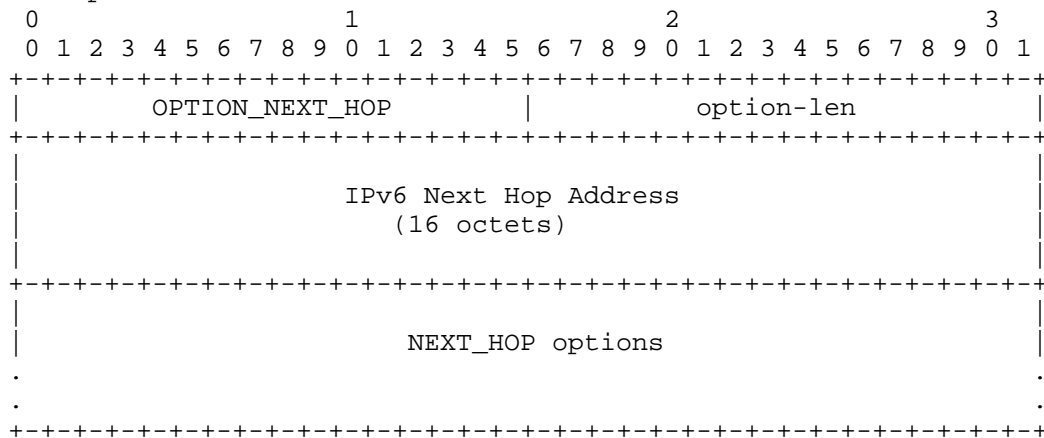


Figure 2: IPv6 Next Hop Option Format

option-code: OPTION\_NEXT\_HOP (TBD1).

option-len: 16 + Length of NEXT\_HOP options field.

IPv6 Next Hop Address: 16 octet long field that specified IPv6 address of the next hop.

NEXT\_HOP options: Options associated with this Next Hop. This includes, but is not limited to, zero, one or more RT\_PREFIX options that specify prefixes reachable through the given next hop.

## 5.2. Route Prefix Option Format

The Route Prefix Option is used to convey information about a single prefix that represents the destination network. The Route Prefix Option is used as a sub-option in the previously defined Next Hop Option. It may also be sent directly in message to indicate that route is available directly on-link.

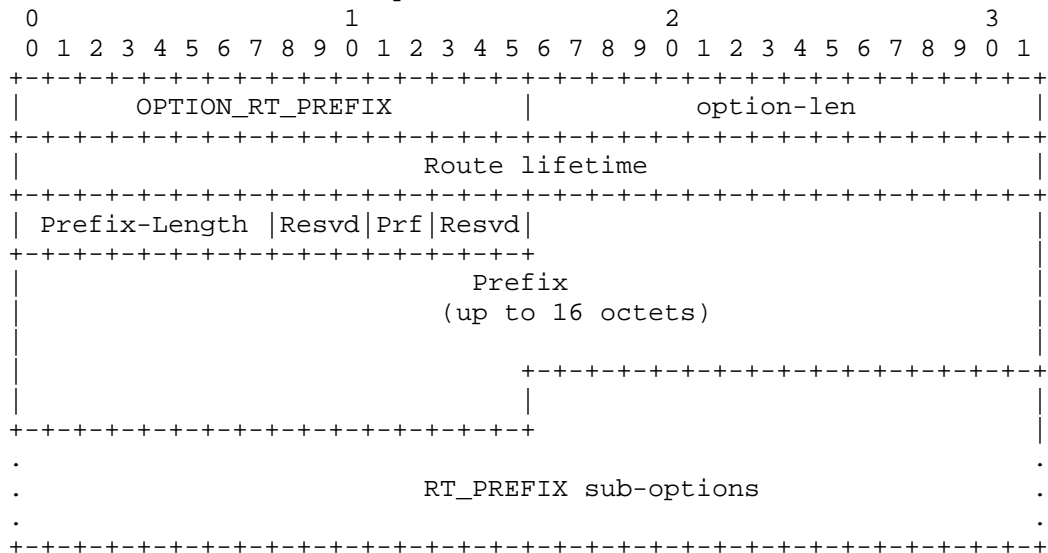


Figure 3: Route Prefix Option Format

option-code: OPTION\_RT\_PREFIX (TBD2).

option-len: Length of the Route Prefix option including all its sub-options.

Route lifetime 32-bit unsigned integer. Specifies lifetime of the route information, expressed in seconds (relative to the time the packet is sent). There are 2 special values defined. 0 means that route is no longer valid and must be removed by clients. A value of all one bits (0xffffffff) represents infinity. means infinity.

Prefix Length: 8-bit unsigned integer. The length in bits of the IP Prefix. The value ranges from 0 to 128. This field represents the number of valid leading bits in the prefix.



- Resvd: Reserved field. Server MUST set this value to zero and client MUST ignore its content.
- Prf(Route Preference): 2-bit signed integer. The Route Preference indicates whether to prefer the router associated with this prefix over others, when multiple identical prefixes (for different routers) have been received. If the Reserved (10) value is received, the Route Information Option MUST be ignored.
- Metric: Route Metric. 8-bit signed integer. The Route Metric indicates whether to prefer the next hop associated with this prefix over others, when multiple identical prefixes (for different next hops) have been received.
- Prefix: a variable size field that specifies Rule IPv6 prefix. Length of the field is defined by prefix6-len field and is rounded up to the nearest octet boundary (if case when prefix6-len is not divisible by 8). In such case additional padding bits must be zeroed.

RT\_PREFIX options: Options specific to this particular prefix.

Values for preference field have meaning identical to Route Information Option, defined in [RFC4191], Section 2.1:

01 High

00 Medium (default)

11 Low

10 Reserved - MUST NOT be sent

## 6. DHCPv6 Server Behavior

When configured to do so, a DHCPv6 server shall provide the Next Hop and Route Prefix Options in ADVERTISE and REPLY messages sent to a client that requested the route option. Each Next Hop Option sent by the server must convey at least one Route Prefix Option.

Server includes NEXT\_HOP option with possible RT\_PREFIX suboptions to designate that specific routes are available via routers. Server includes RT\_PREFIX options directly in Advertise and Reply messages to inform that specific routes are available directly on-link.

If there is more than one route available via specific next hop,

server MUST send only one NEXT\_HOP for that next hop, which contains multiple RT\_PREFIX options. Server MUST NOT send more than one identical (i.e. with equal next hop address field) NEXT\_HOP option.

Servers SHOULD NOT send Route Option to clients that did not explicitly requested it, using the ORO.

Servers MUST NOT send Route Option in messages other than ADVERTISE or REPLY.

Servers MAY also include Status Code Option, defined in Section 22.13 of the [RFC3315] to indicate the status of the operation.

Servers MUST include the Status Code Option, if the requested routing configuration was not successful and SHOULD use status codes as defined in [RFC3315] and [RFC3633].

The maximum number of routing information in one DHCPv6 message depend on the maximum DHCPv6 message size defined in [RFC3315]

## 7. DHCPv6 Client Behavior

A DHCPv6 client compliant with this specification MUST request the NEXT\_HOP and RT\_PREFIX Options in an Option Request Option (ORO) in the following messages: Solicit, Request, Renew, Rebind, and Information-Request. The messages are to be sent as and when specified by [RFC3315].

When processing a received Route Options a client MUST substitute a received 0::0 value in the Next Hop Option with the source IPv6 address of the received DHCPv6 message. It MUST also associate a received Link Local next hop addresses with the interface on which the client received the DHCPv6 message containing the route option. Such a substitution and/or association is useful in cases where the DHCPv6 server operator does not directly know the IPv6 next-hop address, other than knowing it is that of a DHCPv6 relay agent on the client LAN segment. DHCPv6 Packets relayed to the client are sourced by the relay using this relay's IPv6 address, which could be a link local address.

The Client SHOULD refresh assigned route information periodically. The generic DHCPv6 Information Refresh Time Option, as specified in [RFC4242], can be used when it is desired for the client to periodically refresh of route information.

The routes conveyed by the Route Option should be considered as complimentary to any other static route learning and maintenance

mechanism used by, or on the client with one modification: The client MUST flush DHCPv6 installed routes following a link flap event on the DHCPv6 client interface over which the routes were installed. This requirement is necessary to automate the flushing of routes for clients that may move to a different network.

Client MUST confirm that routers announced over DHCPv6 are reachable, using one of methods suitable for specific network type. The most common mechanism is Neighbor Unreachability Detection (NUD), specified in [RFC4861]. Client SHOULD use NUD to verify that received routers are reachable before adjusting its routing tables. Client MAY use other reachability verification mechanisms specific to used network technology. To avoid potential long-lived routing black holes, client MAY periodically confirm that router is still reachable.

#### 7.1. Conflict resolution

Information received via Route Options over DHCPv6 MUST be treated equally to routing information obtained via other sources. In particular, from the RA perspective, DHCPv6 provisioning should be treated as if yet another RA was received. Preference field should be taken into consideration during route information processing. In particular, administrators are encouraged to read [RFC4191], Section 4.1 for guidance.

To facilitate information merge between DHCPv6 and RA, DHCPv6 option conveys the same information as RIO, specified in [RFC4191], albeit on-wire format is slightly different. The differences are:

Metric field (available in previous version of this draft) has been replaced with 2-bit preference field that is in line with RIO information.

RIO uses 128-length prefix field, while DHCPv6 option uses variable prefix length. That difference is used to minimize packet size as it avoid transmitting zeroed octets. Despite slightly different encoding, delivered information is exactly the same.

If prefix is available directly on-link, Route Prefix option is conveyed directly in DHCPv6 message, not withing Next Hop option. That feature is considered a superset, compared to RIO.

#### 8. IANA Considerations

IANA is kindly requested to allocate DHCPv6 option code TBD1 to the OPTION\_NEXT\_HOP and TBD2 to OPTION\_RT\_PREFIX. Both values should be

added to the DHCPv6 option code space defined in Section 24.3 of [RFC3315].

## 9. Security Considerations

The overall security considerations discussed in [RFC3315] apply also to this document. The Route option could be used by malicious parties to misdirect traffic sent by the client either as part of a denial of service or man-in-the-middle attack. An alternative denial of service attack could also be realized by means of using the route option to overflowing any known memory limitations of the client, or to exceed the client's ability to handle the number of next hop addresses.

Neither of the above considerations are new and specific to the proposed route option. The mechanisms identified for securing DHCPv6 as well as reasonable checks performed by client implementations are deemed sufficient in addressing these problems.

It is essential that clients verify that announced routers are indeed reachable, as specified in Section 7. Failing to do so may create black hole routing problem.

This mechanism may introduce severe problems if deployed in networks that use dynamic routing protocols. See Section 4.6 for details.

DHCPv6 becomes a complete provisioning protocol with this mechanism, i.e. all necessary configuration parameters may be delivered using DHCPv6 only. It was suggested that in some cases this may lead to decision of disabling RA. While RA-less networks could offer lower operational expenses and protection against rogue RAs, they would not work with nodes that do not support this feature. Therefore such decision is not recommended, unless all effects are carefully analyzed. It is worth noting that disabling RA support in hosts would solve rogue RA problem, it would in fact only change the issue into rogue DHCPv6 problem. That is somewhat beneficial, however, as rogue RA may affect all nodes immediately while rogue DHCPv6 server will affect only new nodes, that boot up after rogue server manifests itself.

Reader is also encouraged to read DHCPv6 security considerations document [I-D.ietf-dhc-secure-dhcpv6].

## 10. Contributors and Acknowledgements

This document would not have been possible without the significant

contribution provided by: Arifumi Matsumoto, Hui Deng, Richard Johnson, and Zhen Cao.

The authors would also like to thank Alfred Hines, Ralph Droms, Ted Lemon, Ole Troan, Dave Oran, Dave Ward, Joel Halpern, Marcin Siodelski, Alexandru Petrescu, Roberta Maglione, Tim Chown, Brian Carpenter, Dave Thaler, Lorenzo Colitti and Leo Bicknell for their comments and useful suggestions.

This work has been partially supported by Department of Computer Communications (a division of Gdansk University of Technology) and the Polish Ministry of Science and Higher Education under the European Regional Development Fund, Grant No. POIG.01.01.02-00-045/09-00 (Future Internet Engineering Project).

## 11. References

### 11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

### 11.2. Informative References

- [BBF-WT-124]  
Broadband Forum, "BBF WT-124 issue 3", BBF WT-124i3, 2011.
- [I-D.ietf-6man-addr-select-opt]  
Matsumoto, A., Fujisaki, T., Kato, J., and T. Chown,  
"Distributing Address Selection Policy using DHCPv6",  
draft-ietf-6man-addr-select-opt-03 (work in progress),  
February 2012.
- [I-D.ietf-dhc-secure-dhcpv6]  
Jiang, S. and S. Shen, "Secure DHCPv6 Using CGAs",  
draft-ietf-dhc-secure-dhcpv6-04 (work in progress),  
December 2011.
- [I-D.ietf-mif-dns-server-selection]

Savolainen, T., Kato, J., and T. Lemon, "Improved DNS Server Selection for Multi-Interfaced Nodes", draft-ietf-mif-dns-server-selection-07 (work in progress), October 2011.

[I-D.ietf-v6ops-ipv6-multihoming-without-ipv6nat]

Troan, O., Miles, D., Matsushima, S., Okimoto, T., and D. Wing, "IPv6 Multihoming without Network Address Translation", draft-ietf-v6ops-ipv6-multihoming-without-ipv6nat-04 (work in progress), February 2012.

[RFC3442] Lemon, T., Cheshire, S., and B. Volz, "The Classless Static Route Option for Dynamic Host Configuration Protocol (DHCP) version 4", RFC 3442, December 2002.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.

[RFC4242] Venaas, S., Chown, T., and B. Volz, "Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 4242, November 2005.

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

[RFC6204] Singh, H., Beebee, W., Donley, C., Stark, B., and O. Troan, "Basic Requirements for IPv6 Customer Edge Routers", RFC 6204, April 2011.

[THREEGPP-23.853]

Stojanovski, S., "3GPP TR 23.853: Operator Policies for IP Interface Selection (OPIIS)", 3GPP TR 23.853, August 2011, <<http://www.3gpp.org/ftp/Specs/html-info/23853.htm>>.

[nanog-beijnum]

van Beijnum, I., "", , June 2011, <<http://mailman.nanog.org/pipermail/nanog/2011-June/037242.html>>.

## Authors' Addresses

Wojciech Dec  
Cisco Systems  
Haarlerbergweg 13-19  
1101 CH Amsterdam  
The Netherlands

Email: wdec@cisco.com

Tomasz Mrugalski (editor)  
Internet Systems Consortium, Inc.  
950 Charter Street  
Redwood City, CA 94063  
USA

Phone: +1 650 423 1345  
Email: tomasz.mrugalski@gmail.com

Tao Sun  
China Mobile  
Unit2, 28 Xuanwumenxi Ave  
Beijing, Xuanwu District 100053  
China

Phone:  
Email: suntao@chinamobile.com

Behcet Sarikaya  
Huawei USA  
1700 Alma Dr. Suite 500  
Plano, TX 75075  
United States

Phone: +1 972-509-5599  
Fax:  
Email: sarikaya@ieee.org  
URI:





Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: May 17, 2017

G. Chen  
China Mobile  
C. Williams  
Consultant  
D. Wing  
A. Yourtchenko  
Cisco Systems, Inc.  
November 13, 2016

Happy Eyeballs Extension for Multiple Interfaces  
draft-ietf-mif-happy-eyeballs-extension-11

Abstract

This memo proposes extensions to the Happy Eyeball's algorithm requirements defined in RFC6555 for use with the multiple provisioning domain architecture. The Happy Eyeballs in MIF would make the selection process smoother by using connectivity tests over pre-filtered interfaces according to defined policy. This would choose the best interface with an automatic fallback mechanism.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on May 17, 2017.

Copyright Notice

Copyright (c) 2016 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. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	3
3. Use Cases . . . . .	3
3.1. WiFi is broken . . . . .	3
3.2. Policy Conflict . . . . .	4
4. Happiness Parameters . . . . .	4
4.1. Hard Set . . . . .	5
4.1.1. Operator Policy . . . . .	5
4.1.2. User Preference . . . . .	5
4.2. Soft Set . . . . .	6
4.2.1. Provisioning Domain Identity . . . . .	6
4.2.2. DNS Selection . . . . .	6
4.2.3. Next Hop . . . . .	6
4.2.4. Source Address Selection . . . . .	6
4.2.5. Common Practice . . . . .	6
5. HE-MIF Process Requirements . . . . .	7
5.1. First Step, Filter . . . . .	7
5.2. Second Step, Sort . . . . .	8
5.2.1. Interface Validation . . . . .	8
5.2.2. Name Resolution . . . . .	8
5.2.3. Connection Establishment . . . . .	8
6. Implementation Framework . . . . .	9
7. Additional Considerations . . . . .	9
7.1. Usage Scope . . . . .	9
7.2. Fallback Timeout . . . . .	9
7.3. DNS Selections . . . . .	10
7.4. Flow Continuity . . . . .	11
7.5. Interworking with Happy Eyeball . . . . .	11
7.6. Multipath Applicability . . . . .	11
8. IANA Considerations . . . . .	11
9. Security Considerations . . . . .	12
10. Acknowledgements . . . . .	12
11. References . . . . .	12
11.1. Normative References . . . . .	12
11.2. Informative References . . . . .	13
Authors' Addresses . . . . .	14

## 1. Introduction

The MIF problem statement [RFC6418] describes problems specific for nodes attached to multiple provisioning domains. Specifically, there is an issue description that a node has selected an interface and obtained a valid IP address from the network, but Internet connectivity is not available. This memo intends to address the issue and elaborate more in Section 3.1.

[RFC7556] describes the multiple provisioning domain architecture. It refers to using connectivity tests to validate a Provisioning Domain (PvD). Given a number of implicit/explicit PvDs, plus preferences/policy, what is the process to follow to select the best PvD to use for any given connection. In the event that two or more are deemed to be best, how are the Happy Eyeballs (HE) techniques applied to find the best and deal with resilience. This memo also proposes process requirements using Happy Eyeballs (HE) extensions.

There are a variety of algorithms that can be envisioned. This document describes additional parameters and processes that need to be considered in addition to the HE algorithm requirements defined in [RFC6555] necessary to support multiple interfaces, so that a node with multiple interfaces can select the best path for a particular connection-oriented flow (e.g., TCP, SCTP).

## 2. Terminology

This document makes use of following terms:

- o Happy Eyeballs (HE): specifies requirements for an algorithm that reduces the user-visible connection delay for dual-stack hosts with a single interface per-protocol.
- o Happy Eyeballs - Multi-Interface (HE-MIF): Extends the Happy Eyeballs concept to the multiple provisioning domain architecture. It describes additional requirements for algorithms that offer connectivity tests on PVD-aware or non-PVD-aware nodes [RFC7556] to select the best interface for a specific connection request.

## 3. Use Cases

The section describes scenarios the HE-MIF targeted to use.

### 3.1. WiFi is broken

Assuming a MIF node has both a 3GPP mobile network interface and a WiFi interface, a common practice would be to always prefer the WiFi connection when the node enters an area with WiFi available. In this

situation, a node might assume that because a valid IP address has been allocated, the WiFi link provides connectivity to destinations through the Internet. However, this might not be the case for several reasons:

- o WiFi access-point authentication requirements
- o WiFi has no global Internet connectivity
- o Instability at layer 2

In order to resolve this problem, the user would need to disable the device's interface preferences, e.g. by disabling the WiFi interface. HE-MIF offers users the possibility of configuring their preferences for the choice of the most suitable network interface to use, such as via setting on their mobile phone.

In this case, users may prefer to wait an appropriate time period for connections to be established over a WiFi path. If no connection can be made it will fall back to attempting the connection over a 3GPP mobile network path.

### 3.2. Policy Conflict

A node has network access via both WiFi and 3GPP networks. In a mobile network, IPv6-only may be preferable since IPv6 has the potential to be simpler than dual-stack. The WiFi access offers IPv4 only. In this scenario, the combination of source address selection [RFC6724] and preferring the WiFi interface may cause a problem. The transition to IPv6 may mean that IPv6 is the preferred protocol, so the 3GPP interface should be chosen even though it could be considered a suboptimal selection e.g. the WiFi interface likely is less expensive.

## 4. Happiness Parameters

This section provides input parameter proposal that HE-MIF should catch. Two sets of "Happiness" parameters have been defined. It serves applications and initiates HE-MIF connection tests subsequently. By following the process described below, MIF nodes can select an appropriate interface that best meets the configuration parameters defined by the user. The two sets of "Happiness" parameters are called Hard Set and Soft Set respectively.

#### 4.1. Hard Set

Hard set contains parameters which should be complied with. It helps to select candidate interfaces through which a particular flow should be directed. These should be seen as constraints on the choice, such as provider policies, support for IPv4 or IPv6, and other parameters which would prevent a particular interface and transport from being used by a particular flow. Parameters in the hard set should be easy to use and understand. When several parameters in the hard set are in conflict, the user's preference should be prioritized.

##### 4.1.1. Operator Policy

Operators may deliver the customized policies for a particular network environment because of geo-location or service regulation considerations. One example relevant for 3GPP networks is an operator delivering policies from an Access Network Discovery and Selection function (ANDSF) [TS23.402].

The ANDSF provides a node with policies and network selection information to influence the selection between different access technologies, such as 3GPP mobile networks, WiFi access. The ANDSF can provide the node with three types of information[TS24.302].

- o Access network discovery and selection information: it includes a list of access networks available in the vicinity of the node. The information may include the access technology types (e.g. WiFi), network identifiers (e.g. SSID in the case of WiFi) as well as validity conditions (e.g. where and when).
- o Inter-System Mobility Policies (ISMPs): they are a set of operator-defined rules and preferences that affect the inter-system mobility decisions, e.g. decisions about whether to use 3GPP mobile network or a WiFi network.
- o Inter-System Routing Policies (ISRPs): the node uses ISRPs when it can route IP traffic simultaneously over multiple radio access networks. It could provide routing policies in an IP flow granularity.

##### 4.1.2. User Preference

User's preference: users may express preferences which likely not have a formally technical language, like "No 3/4G while roaming", "Only download applications larger than 20Mb over WiFi", etc. Those information are normally input from User Interface (UI).

## 4.2. Soft Set

Soft set contains factors which impact the selection of the path across which a particular flow should be transmitted among the available interfaces and transports which meet the hard set requirements described above.

### 4.2.1. Provisioning Domain Identity

A PVD-aware node uses PVD Identity(PvD-ID) to select a PvD with a matching ID for special-purpose connection requests. The PvD-ID may be generated by the node implicitly or received from the network explicitly. For explicit PvDs, the node could take the parameter from PvD ID Option [I-D.ietf-mif-mpvd-id] via the configuration protocols ([I-D.ietf-mif-mpvd-dhcp-support] or [I-D.ietf-mif-mpvd-ndp-support]). A PVD-aware node may decide to use one preferred PVD or allow the use of multiple PVDs simultaneously for applications. The node behavior should be consistent with MPVD architecture [RFC7556].

### 4.2.2. DNS Selection

At the name service lookup step, the node has to choose a recursive DNS server to use. A HE-MIF node should take the parameter of RDNSS Selection DHCP Option [RFC6731] to select an interface for a particular namespace.

### 4.2.3. Next Hop

[RFC4191] allows the configuration of specific routes to a destination. A HE-MIF node should take the parameters of router preference and route information to identify the next hop.

### 4.2.4. Source Address Selection

For each destination, once the best next hop is found, the node should consider IP prefix and precedence parameter in policy table to select the best source address according to the rule defined in [RFC6724].

### 4.2.5. Common Practice

There is relevant common practice related to interface selection, e.g. Prefer WiFi over a 3GPP interface, if available. Such conventions should also be considered.

## 5. HE-MIF Process Requirements

An HE-MIF node may use the two sets of parameters as two steps in the interface selection process. The first step is to use the Hard Set to synthesize policies from different actors (e.g., users or network operators). These hard set parameters will provide a filter which will exclude not qualifying interfaces from any further consideration.

The second step is to influence how a node makes a connection when multiple interfaces still remain in the candidate list after first step. This is essentially sorting behavior. In the multiple provisioning domain architecture, a PVD aware node makes connectivity tests as described in Section 5.3 of [RFC7556]. A PVD agnostic node take other parameters apart from PVD-ID in the Soft Set to proceed the sort process.

The two steps are described in more details in the following sub-sections. It should be noted that HE-MIF does not prescribe such two-step model. It will be very specific to particular cases and implementations. The two step model mainly describes requirements for how to use the hard/soft set.

### 5.1. First Step, Filter

One goal of the filter is to reconcile multiple selection policies from users or operators. Afterwards, merged demands would be mapped to a set of candidate interfaces, which are judged as qualified.

Decision on the reconciliation of different policies will depend very much on the deployment scenario. An implementation may not be able to determine priority for each policies without explicit configuration provided by users or administrator. For example, an implementation may by default always prefer the WiFi because of cost saving consideration. Whereas, other users may turn off a device's WiFi interface to guarantee use of a 3GPP network interface to assure higher reliability or security.

The decision on mergence of policies may be made by implementations, or by node administrators. However, it's worth to note that a demand from users should be normally considered higher priority than from other actors.

The merged policies serve as a filter which is iterated across the list of available interfaces. Qualified interfaces are selected and the proceed to the second step.

## 5.2. Second Step, Sort

### 5.2.1. Interface Validation

The Sort process aims to select the best interface and provide fallback capacities. As stated in [RFC7556], a PVD-aware node shall perform connectivity tests and, only after validation of the PVD, consider using it to serve application connections requests. In current implementations, some nodes already implement this, e.g., by trying to reach a dedicated web server (see Section 3.1.2 [RFC6419]). If anything is abnormal, it assumes there is a proxy on the path. This status detection is recommended to be used in HE-MIF to detect DNS interception or an HTTP proxy that forces a login or a click-through. Unexamined PVDs or interfaces should be accounted as "unconnected". It should not join the sort process.

### 5.2.2. Name Resolution

Name resolution is executed on the validated interfaces. Before the requests are initiated, it should check if there is a matching PVD ID for the destination name. A PVD agnostic node may request DNS server selection DHCP option [RFC6731] for interface selection guidance. Those information may weight a particular interface to be preferred to others sending resolving requests. If the node can't find useful information in the Soft Set, DNS queries would be sent out on multiple interfaces in parallel to maximize chances for connectivity. Some additional discussions of DNS selection consideration of HE-MIF are described in Section 7.3.

### 5.2.3. Connection Establishment

Once a destination address was resolved, a connection is to be setup. For the given destination address, a PVD-aware node selects a next-hop and source address associated with that PVD in the name resolution process. A PVD agnostic node may receive certain next hop in a RA message [RFC4191], the node selects best source address according to the rules [RFC6724].

The interface identified by the source address should be treated to initiate the connection prior to others. This could avoid thrashing the network, by not making simultaneous connection attempts on multiple interfaces. After making a connection attempt on the preferred pairs and failing to establish a connection within a certain time period (see Section 7.2), a HE-MIF implementation will decide to initiate connection attempt using rest of interfaces in parallel. This fallback consideration will make subsequent connection attempts successful on non-preferable interfaces.



The node would cache information regarding the outcome of each connection attempt. Cache entries would be flushed periodically. A system-defined timeout may take place to age the state. Maximum on the order of 10 minutes defined in [RFC6555] is recommended to keep the interface state changes synchronizing with IP family states.

If there is no specific Soft Set provided, all selected interfaces should be treated equally. For a node implementing multipath transports (for example, Multipath TCP (MPTCP) [RFC6182]), the interfaces could be treated as valid to perform subsequent multipath process, such as starting subflow. A node only supporting single physical transport would initiate on several interface simultaneously. The goal here is to provide the most fast connection for users, by quickly attempting to connect using each candidate interface. Afterwards, the node would do the same caching and flushing process as described above.

## 6. Implementation Framework

The simplest way to implement the processes described in this document is within the application itself. This would not require any specific support from the operating system beyond the commonly available APIs that provide transport service. It could also be implemented using a high-level API approach, linking to the MIF-API [I-D.ietf-mif-api-extension].

## 7. Additional Considerations

### 7.1. Usage Scope

Connection-oriented transports (e.g., TCP, SCTP) are directly applied as scoped in [RFC6555]. For connectionless transport protocols (e.g., UDP), a similar mechanism can be used if the application has request/response semantics. Further investigations are out of the document scope.

### 7.2. Fallback Timeout

When the preferred interface was failed, HE-MIF would trigger a fallback process to start connection initiation on several candidate interfaces. A period of time should be set to invalidate the interface and fallback to others. Aggressive timeouts may achieve quick interface handover, but at the cost of traffic that may be chargeable on certain networks, e.g. the handover from WiFi to 3GPP networks brings a charge to customers. Considering the reasons, it is recommended to prioritize the input from users (e.g., real customers or applications) through user interface. For default-setting on a system, a hard error [RFC1122] in replied ICMP could

serve as a trigger for the fallback process. When the ICMP soft error is present or non-response was received, it's recommended that the timeout should be large enough to allow connection retransmission. [RFC1122] states that such timer must be at least 3 minutes to provide TCP retransmission. However, several minutes delay may not be inappropriate for user experiences. A widespread practice [RFC5461] sets 75 seconds to optimize connection process.

More optimal timer may be expected. The particular setting will be very specific to implementations and cases. The memo didn't try to provide a concrete value because of following concerns.

- o RTT (Round-Trip Time) on different interfaces may vary quite a lot. A particular value of timeout may not accurately help to make a decision that this interface doesn't work at all. On the contrary, it may cause a misjudgment on an interface, which is not very fast. In order to compensate the issues, the timeout setting based on past experiences of a particular interface may help to make a fair decision. Whereas, it's going beyond the capability of Happy Eyeballs [RFC6555]. Therefore, it leaves a particular implementation.
- o In some cases, fast interface may not be treated as "best". For example, an interface could be evaluated in the principle of bandwidth-delay, termed "Bandwidth-Delay-Product". Happy Eyeballs measures only connection speed. That is, how quickly a TCP connection is established. It does not measure bandwidth. If the fallback has to take various factors into account and make a balanced decision, it's better to resort to a specific context and implementation.

### 7.3. DNS Selections

During the Sort process, HE-MIF prioritizes PVD-ID match or [RFC6731] inputs to select a proper server. It could help to address following two cases.

- o A DNS answer may be only valid for a specific provisioning domain, but the DNS resolver may not be aware of that because the DNS reply is not kept with the provisioning from which the answer comes. The situation may become worse if asking internal name with public address response or asking public name with private address answers.
- o Some FQDNs can be resolvable only by sending queries to the right server (e.g., intranet services). Otherwise, a response with NXDOMAIN is replied. Fast response is treated as optimal only if

the record is valid. That may cause messy for data connections, since NXDOMAIN doesn't provide useful information.

HE-MIF can help to solve the issues of DNS interception with captive portal. The DNS server modified and replied the answer with the IP address of captive portal rather than the intended destination address. In those cases, TCP connection may succeed, but Internet connectivity is not available. It results in lack of service unless user has authenticated. HE-MIF recommended using network connectivity status probes to examine a pre-configured URL for detecting DNS interception on the path (see more in Section 5.2). The node will be able to automatically rely upon other interfaces to select right DNS servers by excluding the unexamined interfaces.

#### 7.4. Flow Continuity

[I-D.deng-mif-api-session-continuity-guide] describes session continuity guidance for application developers. The flow continuity topic is beyond this document scope.

#### 7.5. Interworking with Happy Eyeball

HE-MIF process could cooperate with HE [RFC6555]. HE is executed on an interface which is selected to make connection establishment (see Section 5.2.3). for example, a node following PvD policy to pick a interface and make both IPv4/IPv6 connection attempts in consistent with HE requirements. The interface state management in HE-MIF is designed to synchronize with IP family states. It could facilitate the HE executions.

#### 7.6. Multipath Applicability

Some nodes may support transports that provide an abstraction of a single connection, aggregating multiple underlying connections. Multipath TCP (MPTCP) [RFC6182] is an example of such a transport protocol. For connections provided by such transports, a node may leverage the "happiness" parameters and process on the underlying connections. Following the HE-MIF requirements, each connection could be performed consistently with user/operator's preference and corresponding provisioning domain information.

### 8. IANA Considerations

This memo does not include any IANA requests.

## 9. Security Considerations

The security consideration is following the statement in [RFC6555] and [RFC6418].

## 10. Acknowledgements

The authors would like to thank Margaret Wasserman, Hui Deng, Erik Kline, Stuart Cheshire, Teemu Savolainen, Jonne Soininen, Simon Perreault, Zhen Cao, Dmitry Anipko, Ted Lemon, Daniel Migault, Russ White and Bing Liu for their helpful comments.

Many thanks to Ralph Droms, Ian Farrer, Jouni Korhonen, Mirja Khlewind and Suresh Krishnan for their detailed reviews.

## 11. References

### 11.1. Normative References

- [RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <<http://www.rfc-editor.org/info/rfc1122>>.
- [RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, DOI 10.17487/RFC4191, November 2005, <<http://www.rfc-editor.org/info/rfc4191>>.
- [RFC6555] Wing, D. and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts", RFC 6555, DOI 10.17487/RFC6555, April 2012, <<http://www.rfc-editor.org/info/rfc6555>>.
- [RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, DOI 10.17487/RFC6724, September 2012, <<http://www.rfc-editor.org/info/rfc6724>>.
- [RFC6731] Savolainen, T., Kato, J., and T. Lemon, "Improved Recursive DNS Server Selection for Multi-Interfaced Nodes", RFC 6731, DOI 10.17487/RFC6731, December 2012, <<http://www.rfc-editor.org/info/rfc6731>>.
- [TS23.402] 3rd Generation Partnership Project, 3GPP., "Architecture enhancements for non-3GPP accesses v8.8.0", December 2009.

[TS24.302]

3rd Generation Partnership Project, 3GPP., "Access to the 3GPP Evolved Packet Core (EPC) via non-3GPP access networks v14.0.0", June 2016.

## 11.2. Informative References

[I-D.deng-mif-api-session-continuity-guide]

Deng, H., Krishnan, S., Lemon, T., and M. Wasserman, "Guide for application developers on session continuity by using MIF API", draft-deng-mif-api-session-continuity-guide-04 (work in progress), July 2014.

[I-D.ietf-mif-api-extension]

Liu, D., Lemon, T., Ismailov, Y., and Z. Cao, "MIF API consideration", draft-ietf-mif-api-extension-05 (work in progress), February 2014.

[I-D.ietf-mif-mpvd-dhcp-support]

Krishnan, S., Korhonen, J., and S. Bhandari, "Support for multiple provisioning domains in DHCPv6", draft-ietf-mif-mpvd-dhcp-support-02 (work in progress), October 2015.

[I-D.ietf-mif-mpvd-id]

Krishnan, S., Korhonen, J., Bhandari, S., and S. Gundavelli, "Identification of provisioning domains", draft-ietf-mif-mpvd-id-02 (work in progress), October 2015.

[I-D.ietf-mif-mpvd-ndp-support]

Korhonen, J., Krishnan, S., and S. Gundavelli, "Support for multiple provisioning domains in IPv6 Neighbor Discovery Protocol", draft-ietf-mif-mpvd-ndp-support-03 (work in progress), February 2016.

[RFC5461] Gont, F., "TCP's Reaction to Soft Errors", RFC 5461, DOI 10.17487/RFC5461, February 2009, <<http://www.rfc-editor.org/info/rfc5461>>.

[RFC6182] Ford, A., Raiciu, C., Handley, M., Barre, S., and J. Iyengar, "Architectural Guidelines for Multipath TCP Development", RFC 6182, DOI 10.17487/RFC6182, March 2011, <<http://www.rfc-editor.org/info/rfc6182>>.

[RFC6418] Blanchet, M. and P. Seite, "Multiple Interfaces and Provisioning Domains Problem Statement", RFC 6418, DOI 10.17487/RFC6418, November 2011, <<http://www.rfc-editor.org/info/rfc6418>>.

[RFC6419] Wasserman, M. and P. Seite, "Current Practices for Multiple-Interface Hosts", RFC 6419, DOI 10.17487/RFC6419, November 2011, <<http://www.rfc-editor.org/info/rfc6419>>.

[RFC7556] Anipko, D., Ed., "Multiple Provisioning Domain Architecture", RFC 7556, DOI 10.17487/RFC7556, June 2015, <<http://www.rfc-editor.org/info/rfc7556>>.

#### Authors' Addresses

Gang Chen  
China Mobile  
29, Jinrong Avenue  
Xicheng District,  
Beijing 100033  
China

Email: [phdgang@gmail.com](mailto:phdgang@gmail.com), [chengang@chinamobile.com](mailto:chengang@chinamobile.com)

Carl Williams  
Consultant  
El Camino Real  
Palo Alto, CA 94306  
USA

Email: [carlw@mcsr-labs.org](mailto:carlw@mcsr-labs.org)

Dan Wing  
Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134  
USA

Email: [dwing@cisco.com](mailto:dwing@cisco.com)

Andrew Yourtchenko  
Cisco Systems, Inc.  
De Kleetlaan, 7  
Diegem B-1831  
Belgium

Email: [ayourtch@cisco.com](mailto:ayourtch@cisco.com)

MIF Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 30, 2012

D. Migault  
Francetelecom - Orange  
C. Williams  
MCSR Labs  
March 29, 2012

Multiple Interfaces IPsec Security Requirements  
draft-mglt-mif-security-requirements-01.txt

Abstract

ISPs want to take advantage of MIF Transport protocols like SCTP, MPTCP to enhance their End User's experience when the End User has been offloaded on WLAN. In addition, WLAN are untrusted so ISPs MUST Secure at least some of their End Users's communications. For various reasons IPsec is the protocol they choose to secure the communications. Currently, IPsec is not adapted to Multiple Interfaces Environment. IPsec can hardly be configured in a proper way which may result in breaking End Users' communications. At least, it makes it very hard for the End Users to combine Security with MIF enhancements. MOBIKE partly address the problem for a single Interface. This draft provides the problem statement and defines the IPsec Security Requirements for MIF.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on September 30, 2012.

Copyright Notice

Copyright (c) 2012 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. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Requirements notation . . . . .	3
2. Introduction . . . . .	3
3. Problem Statement . . . . .	3
3.1. Adding Interfaces Dynamically . . . . .	3
3.2. Removing Interfaces Dynamically . . . . .	4
3.3. Multihoming . . . . .	5
3.4. Hard Handover Mobility . . . . .	5
3.5. Soft Handover Mobility . . . . .	5
3.6. Selecting Traffic . . . . .	6
3.7. Conclusion . . . . .	6
4. Multiple Interfaces Offload Security Requirements . . . . .	7
5. Position toward MOBIKE . . . . .	9
6. Security Considerations . . . . .	10
7. IANA Considerations . . . . .	10
8. Acknowledgment . . . . .	10
9. Normative References . . . . .	10
Authors' Addresses . . . . .	11



## 1. Requirements notation

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 [RFC2119].

## 2. Introduction

Current Radio Access Network (RAN) infrastructure will not be able to deal with the next future traffic increase. Consequently ISPs are willing to offload the RAN traffic on alternate networks like WLAN. RAN and WLAN have different characteristics, and compared to RAN, WLAN may be untrusted, unreliable and the Network Interface management is performed by the End User (EU). As a consequence, when a EU switches a non-secured communication from RAN to WLAN, it MUST be able to secure it. Then communications on WLAN takes advantage of Multiple Interfaces to enhance the EU experience on WLAN. Thus, such communications MUST have their security appropriately configured to keep the communication secured and avoid that Security breaks the communication.

Section Section 3 describes the Problem Statement: an IPsec secured communication cannot benefit from MIF features. Then, section Section 4 provides the IPsec Security Requirements for Multiple Interfaces, Mobility and Multihoming. Section Section 5 positions MOBIKE [RFC4555] toward the Security Requirements, and provides the additional features MUST be defined for MOBIKE.

## 3. Problem Statement

### 3.1. Adding Interfaces Dynamically

The EU may be connected through multiple WLAN Access Points for bandwidth aggregation. Eventually, splitting flows among various Access Points may also be one way to overcome WLAN Access Points unreliability. The EU may be able to add or remove an Interface on a given communication. Protocols like SCTP or MPTCP have especially been designed for that purpose. In fact, SCTP through AS-CONF message is able to dynamically add Interfaces to a given SCTP association.

When the EU is being offloaded, the communication may be secured with IPsec. In this draft we consider two scenarios: (1) One where the communication is encapsulated to a Security Gateway through multiple IPsec tunnels (one per Interface). This scenarios may not require the Server to see the EU with Multiple Interfaces. (2) The other

scenario considers a communication where the EU is connected via Multiple Interfaces directly to the Server. In that case, the communication is secured with IPsec transport mode. The main motivation for using End-to-End security is to limit Security Gateway latencies and limit the security overhead.

When the nodes discovers a new Interface, we expect that IPsec adds this Interface. From the existing IPsec Security Associations related to the communication, IPsec MUST be able to derive for both the EU and the Server the IPsec configuration for the ADDED Interface. More specifically, if the EU is connected to a Security Gateway, the EU MUST configure a new IPsec Tunnel so that the communication can be tunnelled from the new Interface to the Security Gateway. With communication, we mean that the EU may send or receive packets related to the communication. If the EU is directly connected to the Server, the EU MUST configure IPsec so that the communication can be also protected by using the new Interface. Note that IPsec does not define which interface SHOULD be used. IPsec is configured so that other protocols in charge of carrying the traffic may be able to choose one or the other Interface.

Currently IPsec does not provide such mechanisms. This means that any time the EU discovers an Interface, it will have to initiate an IKEv2 negotiation that authenticates the EU and the Server and derives the key material. We want to avoid multiple negotiations for a given communication.

An alternative would be to use MOBIKE Multihoming, which provides the opportunity to the EU to add the new Interface with the ADDITIONAL\_IP\*\_ADDRESS Notify Payload. This would make the new Interface being considered as an Alternate Interface. In other words, this Interface could be used only if the EU would become unreachable on the running Interface. This does not provide Multiple Interfaces. A single Interface is used at a time, and this is what MOBIKE has been designed for. Furthermore MOBIKE only considers the Tunnel mode, which would only address the Security Gateway scenario.

### 3.2. Removing Interfaces Dynamically

The EU may use Multiple connections on WLAN, section Section 3.1 explains why the EU may be able to dynamically ADD interfaces to a given communication. Similarly, this section shows that the EU MUST also be able to REMOVE Interfaces from a communication. There may be multiple reasons to REMOVE an Interface. The Interface may not be reachable, the EU may not want to use this Interface anymore... On a security point of view, when an Interface is not used for a secure communication, IPsec MUST explicitly DISCARD all traffic on that Interface.

Currently IKEv2 provides the possibility to DELETE a Security Association. However, this requires a per Security Association Negotiation. With frequent Interface changes, and the Multiple Interfaces of the EU, this negotiations require too many Notify Payload. The EU, simply wants to advertise the Server to REMOVE an Interface with a single Notify Payload.

MOBIKE overcomes this management issue by using a single Interface. Consequently there is only one active Interface.

### 3.3. Multihoming

Multihoming is the ability to provision Interfaces in case the running Interface is not reachable anymore. For a secure communication, the EU wants to provide one or a range of Alternate IP addresses that MUST be used in case the Primary Interface is not reachable. The difference with ADDing an interface to an given communication is that with Multihoming the Alternate MUST be used only if the Primary Interface is not reachable. On an IPsec point of view, it means that IPsec MUST be configured to DISCARD any packets of the communication unless the Primary Interface is not reachable. When the Primary Interface is not reachable, then IPsec MUST be configured to PROTECT or BYPASS the traffic for the given communication.

Currently MOBIKE provides Multihoming. However, MOBIKE does not make possible to assign a list of Alternate Interfaces to a specific communication. The reason is that MOBIKE only considers a single working interface.

### 3.4. Hard Handover Mobility

Hard Handover Mobility is the ability for a host to update an Interface with another. This generates the packets of the Network to be discarded. In an IPsec point of view, updating the Security Association results in DISCARDing packets sent or received on the new Interface, and accepting (BYPASSing or PROTECTing) packets on the old Interface not anymore used.

IPsec with MOBIKE provides this facility. However, it is only provided for the Tunnel mode.

### 3.5. Soft Handover Mobility

Soft Handover is the ability to switch from an old Interface to the a new Interface with a state where both old and new Interfaces can send or receive traffic so to avoid loosing the packets in the network. Soft Handover can be done with a combination of ADD and REMOVE

operations described in section Section 3.1 and section Section 3.2

As mentioned in section Section 3.1 and section Section 3.2, they are currently NOT handled by MOBIKE.

### 3.6. Selecting Traffic

The EU MUST be able to ADD / REMOVE an Interface, to provide Alternate Interface for Multihoming, or perform some Mobility with Soft Handover or Hard Handover. However in the previous sections such operations have been considered as a global policy for the EU. In fact the EU may not have the same policy for all its traffic. Thus such operations MUST be provided for a given traffic. Motivations may be that the EU may keep some corporate traffic inside a corporate network (private IP addresses, confidentiality...) whereas Internet traffic can use any Interface and especially the one providing the highest bandwidth.

MOBIKE does not provide this kind of facility since it considered a single Interface in use.

### 3.7. Conclusion

This section address common scenario for an EU being offloaded on the a WLAN. The EU may be connected to a Security Gateway or directly connected to the Service. In both cases, the EU MUST be able to:

- ADD an Interface: When the EU has discovered a new Interface, it MUST be able to add this Interface to its current configuration. This means, that IPsec MUST be configured to be able to receive or send traffic on all its interfaces.
- REMOVE an Interface: When the EU notice that one Interface is not active, it MUST be able to remove this Interface to its current configuration. This means that IPsec MUST NOT PROTECT any traffic on this Interface.
- Mobility: The EU MUST be able to perform Hard Handover as well as Soft Handover.
- Multihoming: When one link fails, the EU MUST be able to automatically switch the traffic to an Alternate IP address. This means that IPsec MUST be configured to be able to receive or send traffic on that Interface.
- Traffic Selectors: The EU MUST be able to perform all the above operations globally or for a given traffic. Thus, it MUST be able to indicate which traffic the operation MUST be applied to.

#### 4. Multiple Interfaces Offload Security Requirements

Then follows the Multiple Interfaces Offload Security Requirements. Note they only concern the Security layer. The only purpose of those Requirements is to properly configure the EU Security Layer so that the Security Layer does not stall or affect the EU communication. Since this draft considers IPsec [RFC4301] and IKEv2 [RFC5998], Multiple Interfaces, Multihoming and Mobility address two different channels:

- The DATA channel: i.e. EU communication. In that case, Security Requirements means how to secure properly the IPsec Security Policy Database and Security Association Database, so that IPsec do not block the EU communication. This is like configuring a firewall.
- IKEv2 channel i.e. IKEv2 application. IKEv2 is the IPsec application that configures the IPsec Databases. The application MUST be Multiple Interfaces, Multihoming and Mobility aware so to configure properly the IPsec Databases for the DATA channel.

Here are the following Security Requirements:

- Multiple Interfaces:
  - DATA channel: For the DATA channel, Multiple Interfaces means that the EU MUST be able to ADD or REMOVE an IP address to a given secured communication. Suppose an EU has established a communication with a Server using an Interface I\_OLD. When it detects an new Interface I\_NEW, the EU MUST be able to configure IPsec Databases so that the communication can go through I\_OLD or I\_NEW without being discarded. Note that how the DATA traffic is handled and effectively routed on one or the other or both Interfaces is out of scope of the draft. Similarly, when the EU is communicating to the Server with Multiple Interfaces, it MUST be able to configure IPsec Databases so that one or multiple interfaces MUST NOT accept / handle any traffic.
  - IKEv2 channel: For the IKEv2 channel, we suppose using one interface is sufficient. The IKEv2 channel only carries signalization messages. If the EU wants to change the Interface for IKEv2, then it SHOULD perform a Mobility.
- Multihoming:
  - DATA channel: For the DATA channel, Multihoming means that the EU MUST be able to provide Alternate Interfaces to the Server. In the case the Primary (or running) Interface fails, the communication with the Server MUST be able to go on on the Alternate Interface. More specifically, this means that when the Primary Interface is detected as being down, the EU and the Server MUST

configure the IPsec Databases so that the communication can use the Alternate Interface. The difference with ADDing and Interface in the Multiple Interfaces case is that until the Primary Interface is down, the Alternate Interface does not receive or transmit any traffic. Alternate Interfaces DISCARD such traffic.

- IKEv2 channel: For the IKEv2 channel, Multihoming means that when the Primary Interface is down, IKEv2 MUST be able to switch to the Alternate Interface to send IKEv2 signalization messages to the Server. Once IKEv2 has recovered from the Primary Interface crash-down, it can proceed to the DATA channel IPsec configuration.
- Mobility:
  - DATA channel: For the DATA channel, Mobility means that the EU MUST be able to UPDATE the IPsec Databases and change an old Interface (I\_OLD) by a new Interface (I\_NEW). There are two ways to do so. With a Hard Handover, I\_OLD is replaced by I\_NEW. Packets that are in the network or in the network stack of the Server and EU when the update occurs will be DISCARDED by the EU. With Soft Handover, the EU ADDs I\_NEW and configures its IPsec Databases to receive / send traffic on both I\_OLD and I\_NEW. Then it REMOVES I\_OLD when no traffic is anymore expected on that Interface. Note that Soft Handover is performed according to the Multiple Interfaces Requirements.
  - IKEv2 channel: For the IKEv2 channel, as mentioned in the Multiple Interfaces item, Hard Handover may be sufficient, since the channel only carries signalization messages. Once IKEv2 has moved the IKEv2 channel, it configures IPsec Databases for the DATA channel.
- Traffic Selector:
  - DATA channel: For DATA channel Traffic Selector MUST specify which traffic the Mobility, Multihoming, Multiple Interface action MUST be performed.
  - IKEv2 channel: For the IKEv2, Mobility and Multiple Interface operation may be done with a Hard Handover. However, for Multihoming the channel SHOULD be consider as a specific traffic.

Note that when this draft considers Mobility, Multiple Interfaces or Mobility, only the IPsec configuration is affected. However, in some cases, the configuration of the IPsec Databases may affect the communication of the EU. In fact, if the EU is securing its communication with IPsec and the Tunnel mode, a modification of the outer Interface results in moving the communication. In that case, communication mobility results as a side effect of IPsec Database configuration and this is what is used in MOBIKE [RFC4555]. This case does not happen with the IPsec Transport mode, and the

communication mobility MUST be handled by other protocols than IPsec (application, SHIM6, SCTP, MPTCP...

## 5. Position toward MOBIKE

Multihoming Security Requirements are partly handled by IPsec MOBIKE [RFC4555] extension. MOBIKE has been designed for the VPN Mobility and Multihoming use case with a single interface. Thus MOBIKE only addresses the Security Gateway, with the IPsec Tunnel mode. More specifically, MOBIKE does neither address the Transport mode, nor the case of Multiple Interfaces.

Here are the Mobility and Multihoming MOBIKE features:

- MOBIKE Mobility: MOBIKE provides Mobility by UPDATING the outer IP address. Because MOBIKE considers a single interface, the UPDATE occurs for both the IKEv2 channel and the DATA channel. Furthermore, Because MOBIKE only considers the Tunnel mode, UPDATING the IPsec Databases results in moving the communication as a side effect. Because the EU has a single interface, Mobility is always a Hard Handover.
- MOBIKE Multihoming: MOBIKE provides Multihoming mechanism. The two peers are able to exchange Alternate IP addresses. In case the the Primary IP address is not reachable, IKEv2 tests the Alternate IP address is still reachable with a COOKIE2 exchange. If the Alternate IP address is still reachable, MOBIKE triggers a MOBILITY and UPDATES the Primary Address by the Alternate IP address. Because the EU has only a single interface, both DATA and IKEv2 channels are updated. Because MOBIKE only considers the Tunnel mode, only communications with Tunnel mode will be updated.

MOBIKE provides Mobility and Multihoming features. However, MOBIKE currently partly addresses the Security Requirements:

- Multiple Interfaces: This is NOT addressed by MOBIKE. This means that currently EU with communications involving Multiple Interfaces will need to establish an IKE channel on each Interface. This also means that there is no Security Interface Management facilities, and for example Soft Handover is NOT possible.
- Mobility: MOBIKE addresses Mobility only for Hard Handover with IPsec Tunnel mode protection. As a result the Security Gateway Scenario is partly addressed. To completely address it with Soft Handover, MOBIKE needs to be extended for Multiple Interfaces. Furthermore, to address End-to-End security with the Server, MOBIKE also needs to be extended for the Transport mode.

- Multihoming: MOBIKE Multihoming features currently address the Security Requirements at least for the IKEv2 channel. For the DATA channel, Multihoming may be extended for Multiple Interface by providing Alternate IP addresses for each Interface.

As a result, MOBIKE requires the following extensions:

- Mobility for Transport: to support all offload architecture, especially those with End-to-End Security.
- Mobility for Soft Handover: to make possible Soft Handover for both Transport and Tunnel mode. Note that Soft Handover is related to Multiple Interfaces Management.
- Multihoming for Multiple Interfaces: Multihoming SHOULD be provided with different Alternate IP addresses depending on the network the connection is currently working. Note that it is also related to Multiple Interface Management.
- Multiple Interfaces Management: MOBIKE MUST consider Multiple Interfaces Management for operations it has been designed for like Mobility and Multihoming. It MUST also provide generic extension to make Multiple Interface Management, such as ADDing or REMOVing an Interface.
- Traffic Selector: the EU MUST be able to explicitly specify which traffic the operation applies.

## 6. Security Considerations

The whole draft is about security.

## 7. IANA Considerations

There is no IANA consideration here.

## 8. Acknowledgment

We would like to thank Daniel Palomares, Pierrick Seite, Brian Carpenter, Hui Deng and Jong-Hyouk Lee for their useful comments.

## 9. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.



- [RFC4555] Eronen, P., "IKEv2 Mobility and Multihoming Protocol (MOBIKE)", RFC 4555, June 2006.
- [RFC5998] Eronen, P., Tschofenig, H., and Y. Sheffer, "An Extension for EAP-Only Authentication in IKEv2", RFC 5998, September 2010.

#### Authors' Addresses

Daniel Migault  
Francetelecom - Orange  
38 rue du General Leclerc  
92794 Issy-les-Moulineaux Cedex 9  
France

Phone: +33 1 45 29 60 52  
Email: mglt.ietf@gmail.com

Carl Williams  
MCSR Labs  
Philadelphia, PA 19103  
USA

Phone: 650-279-5903  
Email: carlw@mcsr-labs.org



Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 11, 2013

B. Sarikaya  
Huawei USA  
July 10, 2012

IPv6 RA Options for Multiple Interface Next Hop Routes  
draft-sarikaya-mif-6man-ra-route-01

Abstract

This draft defines new Router Advertisement options for configuring next hop routes on the mobile or fixed nodes. Using these options, an operator can easily configure nodes with multiple interfaces (or otherwise multi-homed) to enable them to select the routes to a destination. Each option is defined together with definitions of host and router behaviors.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on January 11, 2013.

Copyright Notice

Copyright (c) 2012 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. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as

described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	3
3. Default Route Configuration . . . . .	4
4. Route Prefix option . . . . .	4
5. Next Hop Address option . . . . .	5
6. Next Hop Address with Route Prefix option . . . . .	6
7. Security Considerations . . . . .	6
8. IANA Considerations . . . . .	6
9. Acknowledgements . . . . .	7
10. References . . . . .	8
10.1. Normative References . . . . .	8
10.2. Informative References . . . . .	8
Author's Address . . . . .	9

## 1. Introduction

IPv6 Neighbor Discovery and IPv6 Stateless Address Autoconfiguration protocols can be used to configure fixed and mobile nodes with various parameters related to addressing and routing [RFC4861], [RFC4862], [RFC4191]. DNS Recursive Server Addresses and Domain Name Search Lists are additional parameters that can be configured using router advertisements [RFC6106].

Router Advertisements can also be used to configure fixed and mobile nodes in multi-homed scenarios with route information and next hop address. Different scenarios exist such as the node is simultaneously connected to multiple access network of e.g. WiFi and 3G. The node may also be connected to more than one gateway. Such connectivity may be realized by means of dedicated physical or logical links that may also be shared with other users nodes such as in residential access networks.

## 2. Terminology

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 [RFC2119].

### 3. Default Route Configuration

A host, usually a mobile host interested in obtaining routing information usually send a Router Solicitation (RS) message on the link. The router, when configured to do so, provides the route information using zero, one or more Next Hop Address and Route Information options in the router advertisement (RA) messages sent in response.

The route options are extensible, as well as convey detailed information for routes. The router sends one or more Next Hop Address options that specify the IPv6 next hop addresses. Each Next Hop Address option may be associated with zero, one or more Route Prefix options that represent the IPv6 destination prefixes reachable via the given next hop. Router includes Route Prefix option directly in message to indicate that given prefix is available directly on-link. Router MAY send a single Next Hop Address without any Route Prefix options. When router sends Next Hop Address option that is associated with Router Prefix option, the router MUST use Next Hop and Route Prefix option defined in Section 6. The Route Prefix MAY contain `::/0`, i.e. with Prefix Length set to zero to indicate available default route.

RS and RA exchange is for next hop address and route information determination and not for determining the link-layer address of the router. Subsequent Neighbor Solicitation and Neighbor Advertisement exchange can be used to determine link-layer address of the router.

It should be noted that the proposed options in this document will need a central site-wide configuration mechanism. The required values can not automatically be derived from routing tables.

### 4. Route Prefix option

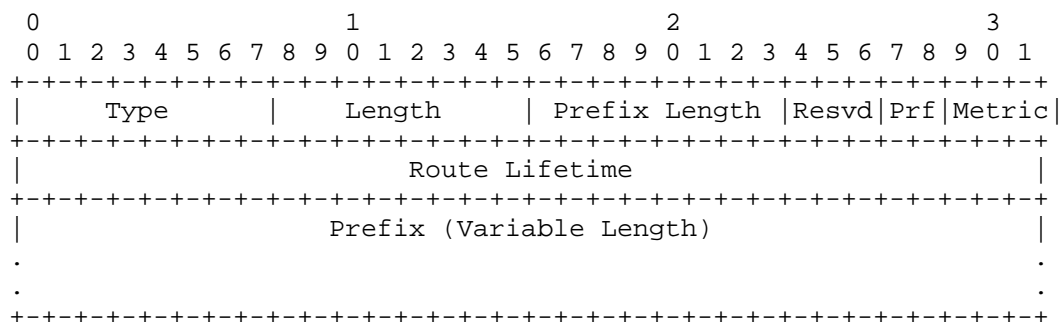


Figure 1: Route Prefix option

Fields:

Type: TBD.

Length: The length of the option (including the Type and Length fields) in units of 8 octets.

Other fields are as in [RFC4191] except:

Metric Route Metric. 3-bit signed integer. The Route Metric indicates whether to prefer the next hop associated with this prefix over others, when multiple identical prefixes (for different next hops) have been received.

#### 5. Next Hop Address option

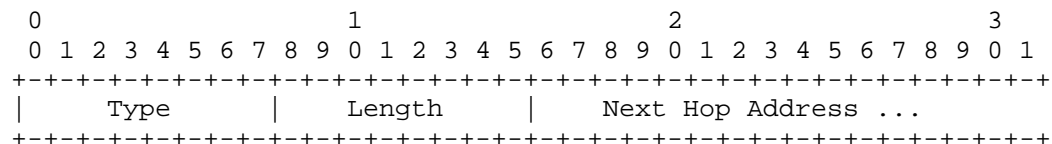


Figure 2: Next Hop Address option

Fields:

Type: TBD.

Length: The length of the option (including the type and length fields) in units of 8 octets. It's value is 3.

Next Hop Address: An IPv6 address that specifies IPv6 address of the next hop. It is 16 octets in length.

## 6. Next Hop Address with Route Prefix option

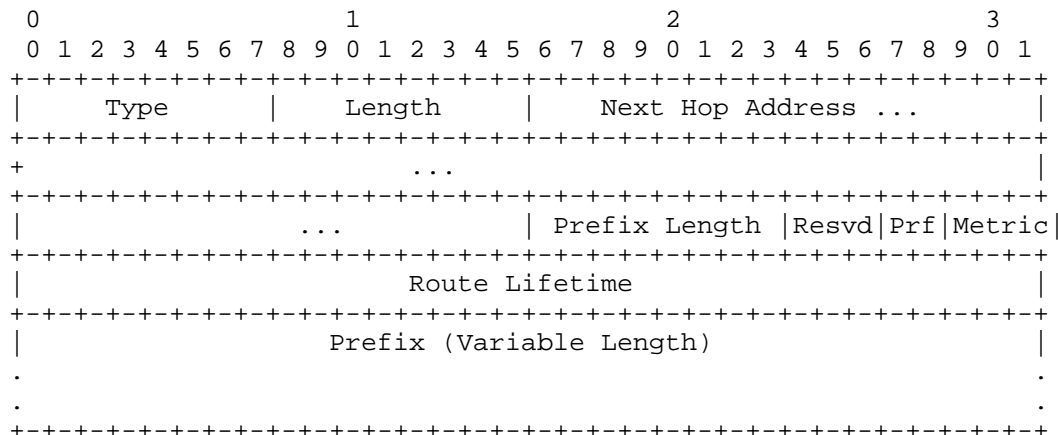


Figure 3: Next Hop Address with Route Prefix option

Fields:

Type: TBD.

Length: The length of the option (including the type and length fields) in units of 8 octets. For example, the length for a prefix of length 16 is 5.

Other fields are as in Section 4 and Section 5.

## 7. Security Considerations

Neighbor Discovery is subject to attacks that cause IP packets to flow to unexpected places. Because of this, neighbor discovery messages MUST be secured, possibly using Secure Neighbor Discovery (SEND) protocol [RFC3971].

## 8. IANA Considerations

Authors of this document request IANA to assign three new RA options:



Option Name	Type
Route Prefix	
Next Hop Address	
Next Hop Address and Route Prefix	

Table 1:

## 9. Acknowledgements

Brian Carpenter provided comments that have led to improvements in the document. We are also grateful to Zhen Cao for his comments.

## 10. References

### 10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.
- [RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.
- [RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.
- [RFC4605] Fenner, B., He, H., Haberman, B., and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, August 2006.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

### 10.2. Informative References

- [RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, November 2010.

Author's Address

Behcet Sarikaya  
Huawei USA  
5340 Legacy Dr. Building 175  
Plano, TX 75024

Phone:  
Email: sarikaya@ieee.org

