

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: April 23, 2012

S. Bhandari
G. Halwasia
S. Bandi
S. Gundavelli
Cisco Systems
H. Deng
China Mobile
October 21, 2011

DHCPv6 class based prefix
draft-bhandari-dhc-class-based-prefix-00

Abstract

DHCPv6 defines class based allocation of IA_NA and IA_TA IPv6 addresses. This document extends DHCPv6 prefix delegation with class based prefix allocation. It defines a new prefix class option to classify a prefix. It defines the behavior of a DHCPv6 client requesting a prefix to include the class of the prefix to be allocated and the DHCPv6 server behavior to select and offer a prefix from a given class. It discusses how IA_NA can be requested and assigned from a specific prefix class.

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 April 23, 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	3
1.1. Motivation	3
1.1.1. Mobile network	3
1.1.2. Homenet	4
1.2. Terminology	5
1.3. Requirements Language	5
2. Overview	5
2.1. Prefix Class Option in IA_PD	6
2.2. Consideration for different DHCPv6 entities	6
2.2.1. Requesting Router Behavior	7
2.2.2. Delegating Router Behavior	7
2.2.3. DHCPv6 Client Behavior for IA_NA allocation	8
2.3. Usage	8
2.3.1. Class based prefix and IA_NA allocation	8
2.3.2. Class based prefix and IA_PD allocation	9
2.3.3. Class based prefix and SLAAC	9
3. Example Application	9
3.1. Class based prefix delegation	10
3.2. IPv6 address assignment from class based prefix	11
3.3. IPv6 prefix delegation from class based prefix	12
4. Acknowledgements	12
5. IANA Considerations	12
6. Security Considerations	12
7. References	12
7.1. Normative References	12
7.2. Informative References	13
Authors' Addresses	13

1. Introduction

DHCPv6 based prefix delegation as defined in [RFC3633] is a mechanism for the delegation of IPv6 prefixes using DHCPv6 options. Through these options, a delegating router can delegate prefixes to authorized requesting routers. If the requesting router has to function as a DHCPv6 server there needs to be additional information in the delegated prefix that helps the requesting router to select the address allocation for the DHCPv6 client it serves, from one of the available delegated prefixes.

One way to select an address or longer prefix (from a delegated prefix) to be allocated by a requesting router playing the role of a DHCPv6 server is by introducing additional options in IA_PD to be matched with options for address selection in the DHCPv6 SOLICIT message. [RFC3315] defines the OPTION_USER_CLASS option which is used for selecting address for assignment. This document introduces OPTION_PREFIX_CLASS option in IA_PD option for the purpose of selecting a prefix for further delegation either via IA_NA or IA_PD DHCPv6 request. It defines the behavior of the DHCPv6 server, the DHCPv6 prefix requesting router and the DHCPv6 client to use this option.

1.1. Motivation

In this section motivation for class based prefix delegation that qualifies the delegated prefix with additional class information is described in the context of mobile networks and homenet. The class information attached to a delegated prefix helps to distinguish property of a delegated IPv6 prefix and selection of the prefix by different applications using it.

1.1.1. Mobile network

In the mobile network architecture, there is a mobile router which functions as a IP network gateway and provides IP connectivity to mobile nodes. Mobile router can be the requesting router requesting delegated IPv6 prefix using DHCPv6. Mobile router can assume the role of DHCPv6 server for mobile nodes(DHCPv6 clients) attached to it. A mobile node in mobile network architecture can be associated with multiple IPv6 prefixes belonging to different domains for e.g. home address prefix, care of address prefix as specified in [RFC3775]. The delegated prefixes when seen from the mobile router perspective appear to be like any other prefix, but each prefixes have different properties. Some delegated prefixes may be topologically local and some may be remote prefixes anchored on a global anchor, but available to the local anchor by means of tunneling setup in the network between the local and global anchor.

Some may be local with low latency characteristics suitable for voice call break-out, some may have global mobility support. So, the prefixes have different properties and it is required for the application using the prefix to learn about this property in order to use it intelligently. There is currently no semantics in DHCPv6 prefix delegation that can carry this information to specify properties of a delegated prefix. In this scenario, the mobile router is unable to further delegate a longer prefix intelligently based on properties of the prefix learnt.

1.1.2. Homenet

With the introduction of IPv6 and possible absence of Network Address Translation(NAT) in home networks, the IPv6 source address of the hosts can be used as a parameter for route decision and providing differentiated service for different classes of devices within a home network. [I-D.baker-fun-routing-class] and [I-D.baker-fun-multi-router] introduce use-cases and requirements for source based routing. The home network architecture and associated requirements are specified in [I-D.chown-homenet-arch]. To support source based routing it is necessary to have a mechanism to assign the source address or prefix based on parameters that identify the class of device or network.

[RFC3315] defines OPTION_USER_CLASS option in the IA_NA/IA_TA assignment, which influences the address allocated based on the user class of the device requesting IA_NA or IA_TA. A typical deployment in a home network is the Customer Premise Equipment (CPE) to be a DHCPv6 client requesting a prefix as defined in [RFC3633] from upstream the DHCPv6 server and playing the role of a DHCPv6 server for devices in the Local Area Network (LAN). The CPE can get a shorter prefix from a DHCPv6 server in Wide Area Network(WAN) and allocate longer prefixes to its DHCPv6 clients. Today the CPE has to be manually configured to associate a prefix acquired from the WAN to devices in the LAN. A means of classifying and associating an acquired prefix via DHCPv6 for further delegation either via IA_NA/IA_TA or IA_PD requests is missing.

For e.g. as shown in Figure 1 the CPE in a home network may request prefixes from the DHCPv6 server of the service provider and assume the role of a DHCPv6 server for devices within the home network. Residential and Small-Office/Home-Office (SOHO) networks may have separate domains for their "data network" and "home video network". Devices in these different domains are to be assigned addresses from different prefix ranges. The CPE router will need a way to assign prefixes to the home video network from a prefix that is meant for home video devices to provide differentiated service for such devices in the provider network that has source address based routing policy

configured.

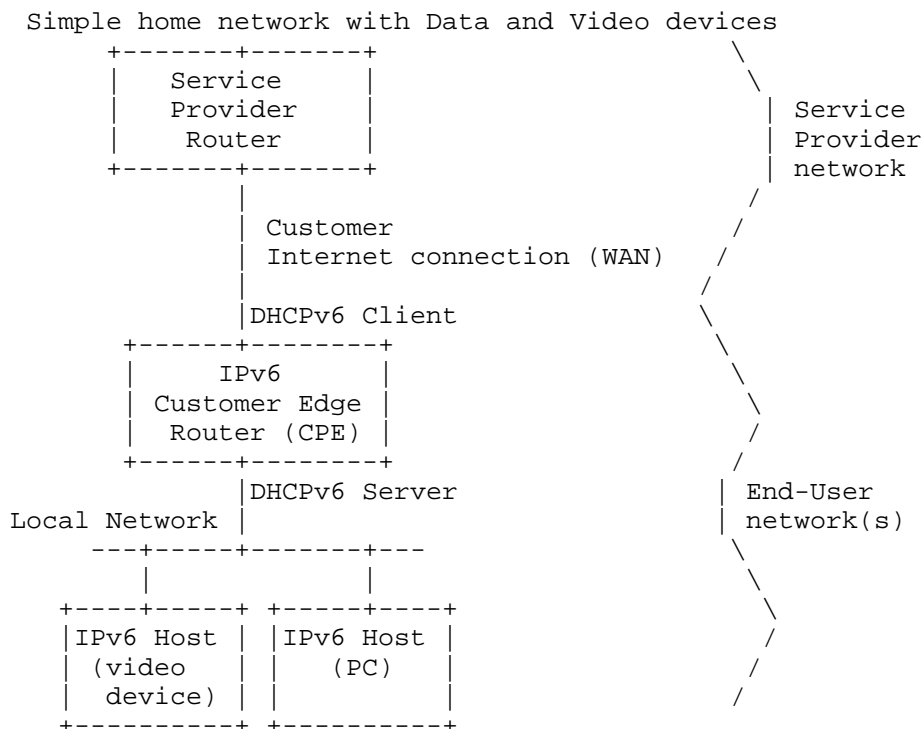


Figure 1

1.2. Terminology

This document uses the terminology defined in [RFC2460], [RFC3315] and [RFC3633].

1.3. Requirements Language

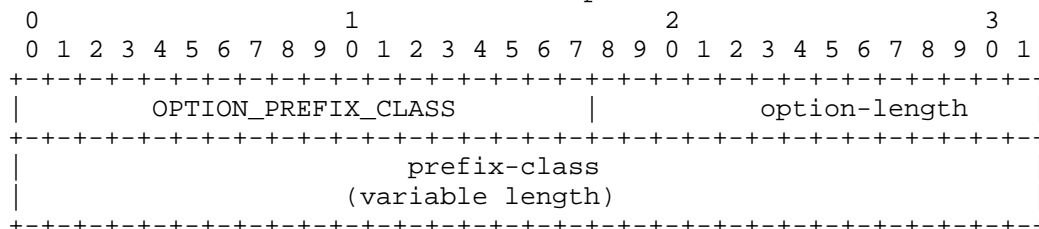
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Overview

This section defines Prefix Class option in IA_PD and IA_NA to aid class based prefix delegation and address assignment. This section defines the behavior of the delegating router, the requesting router and the DHCPv6 client.

2.1. Prefix Class Option in IA_PD

The format of the DHCPv6 Prefix Class option is shown below.



```

option-code:    OPTION_PREFIX_CLASS (TBD)
option-length:  length of prefix-class
prefix-class:   Prefix class (binary string).
    
```

2.2. Consideration for different DHCPv6 entities

The model of operation of communicating prefixes to be used by a DHCPv6 server is as follows. A requesting router requests prefix(es) from the delegating router, as described in Section 2.2.1. A delegating router is provided IPv6 prefixes to be delegated to the requesting router. Examples of ways in which the delegating router is provided these prefixes are:

- o Configuration
- o Prefix delegated via a DHCPv6 request to another DHCPv6 server
- o Using a Authentication Authorization Accounting (AAA) protocol like RADIUS [RFC2865]

The delegating router chooses prefix(es) for delegation, and responds with prefix(es) to the requesting router along with additional options in the allocated prefix as described in Section 2.2.2. The requesting router is then responsible for the delegated prefix(es) after the DHCPv6 REQUEST message exchange. For example, the requesting router may create DHCPv6 server configuration pools from the delegated prefix, and function as a DHCPv6 Server. When the requesting router then receives a DHCPv6 IA_NA requests it can select the address to be allocated based on the OPTION_USER_CLASS or OPTION_PREFIX_CLASS options received in IA_NA request or any of the other methods as described in Section 2.3.1.

2.2.1. Requesting Router Behavior

DHCPv6 requesting router can request for prefixes in the following ways:

- o In the SOLICIT message within the IA_PD Prefix option, it MAY include OPTION_PREFIX_CLASS requesting prefix delegation for the specific class indicated in the OPTION_PREFIX_CLASS option. It can include multiple IA_PD Prefix options to indicate it's preference for more than one prefix class.
- o In the SOLICIT message include an OPTION_ORO option with the OPTION_PREFIX_CLASS option code to request prefixes from all the classes that the DHCPv6 server can provide to this requesting Router.

The requesting router parses the OPTION_PREFIX_CLASS option in the OPTION_IAPREFIX option area of the corresponding IA_PD Prefix option in the ADVERTISE message. The Requesting router MUST then include all or subset of the received class based prefix(es) in the REQUEST message so that it will be responsible for the prefixes selected.

2.2.2. Delegating Router Behavior

If the Delegating router supports class based prefix allocation by supporting the OPTION_PREFIX_CLASS option and it is configured to assign prefixes from different classes, it selects prefixes for class based prefix allocation in the following way:

- o If requesting router includes OPTION_PREFIX_CLASS within the IA_PD Prefix option, it selects prefixes to be offered from that specific class.
- o If requesting router includes OPTION_PREFIX_CLASS within OPTION_ORO, then based on its configuration and policy it MAY offer prefixes from multiple classes available.

The delegating router responds with an ADVERTISE message after populating the IP_PD option with prefixes from different prefix classes. Along with including the IA_PD prefix options in the IA_PD option, it also includes the OPTION_PREFIX_CLASS option in the OPTION_IAPREFIX option area of the corresponding IA_PD prefix option.

If neither the OPTION_ORO nor the IA_PD option in the SOLICIT message include the OPTION_PREFIX_CLASS option, then the delegating router MAY allocate the prefix as specified in [RFC3633] without including the class option in the IA_PD prefix option in the response.

If OPTION_ORO option in the Solicit message includes the OPTION_PREFIX_CLASS option code but the delegating router does not support the solution described in this specification, then the delegating router acts as specified in [RFC3633]. The requesting router MUST in this case also fall back to the behavior specified in [RFC3633].

If both delegating and requesting routers support class-based prefix allocation, but the delegating router cannot offer prefixes for any other reason, it MUST respond to requesting router with appropriate status code as specified in [RFC3633]. For e.g., if no prefixes are available in the specified class then the delegating router MUST include the status code NoPrefixAvail in the response message.

2.2.3. DHCPv6 Client Behavior for IA_NA allocation

DHCPv6 client MAY request for an IA_NA address allocation from a specific prefix class in the following way:

- o In the SOLICIT message within the IA_NA option, it MAY include the OPTION_PREFIX_CLASS requesting address to be allocated from a specific prefix class indicated in that option.

The DHCPv6 server parses OPTION_PREFIX_CLASS option received and includes it in option area of corresponding OPTION_IA_NA in ADVERTISE message.

2.3. Usage

Class based prefix delegation can be used by the requesting router to configure itself as a DHCPv6 server to serve its DHCPv6 clients. It can allocate longer prefixes from a delegated shorter prefix it received, for serving IA_NA and IA_PD requests.

2.3.1. Class based prefix and IA_NA allocation

The requesting router can use the delegated prefix(es) from different classes (for example "video", "guest", "voice" etc), for assigning the IPv6 addresses to the end hosts through DHCPv6 IA_NA based on a preconfigured mapping with OPTION_PREFIX_CLASS option, the following conditions MAY be observed:

- o It MAY have a pre-configured mapping between the prefix class and OPTION_USER_CLASS option received in IA_NA.
- o It MAY match the OPTION_PREFIX_CLASS if the IA_NA request received contains OPTION_PREFIX_CLASS.

- o It MAY map OPTION_PREFIX_CLASS option to the OPTION_USER_CLASS option by string matching of both these option values.
- o It MAY have a pre-configured mapping between the prefix class and the client DUID received in DHCPv6 message.
- o It MAY have a pre-configured mapping between the prefix class and its network interface on which the IA_NA request was received.

The requesting router playing the role of a DHCPv6 server can ADVERTISE IA_NA from a class of prefix(es) thus selected.

2.3.2. Class based prefix and IA_PD allocation

If the requesting router, receives prefix(es) for different classes (for example "video", "guest", "voice" etc), it can use these prefix(es) for assigning the longer IPv6 prefixes to requesting routers it serves through DHCPv6 IA_PD by assuming the role of delegating router, its behavior is explained in Section 2.2.2.

2.3.3. Class based prefix and SLAAC

DHCPv6 IA_NA and IPv6 Stateless Address Autoconfiguration (SLAAC as defined in [RFC4862]) are two ways by IPv6 addresses can be dynamically assigned to end hosts. Making SLAAC class aware is outside the scope of this document.

3. Example Application

The following sub-sections provide examples of class based prefix delegation and how it is used in a home network. Each of the examples will refer to the below network:

The example network consists of an IPv6 video endpoint, IPv6 hosts, and a Smart grid network consisting of IPv6 sensors and a router that supports Smart Grid Energy Services Interface (ESI) to which sensors are connected. The customer edge router acts as a home gateway router for all the devices and networks within the home.

Example home network

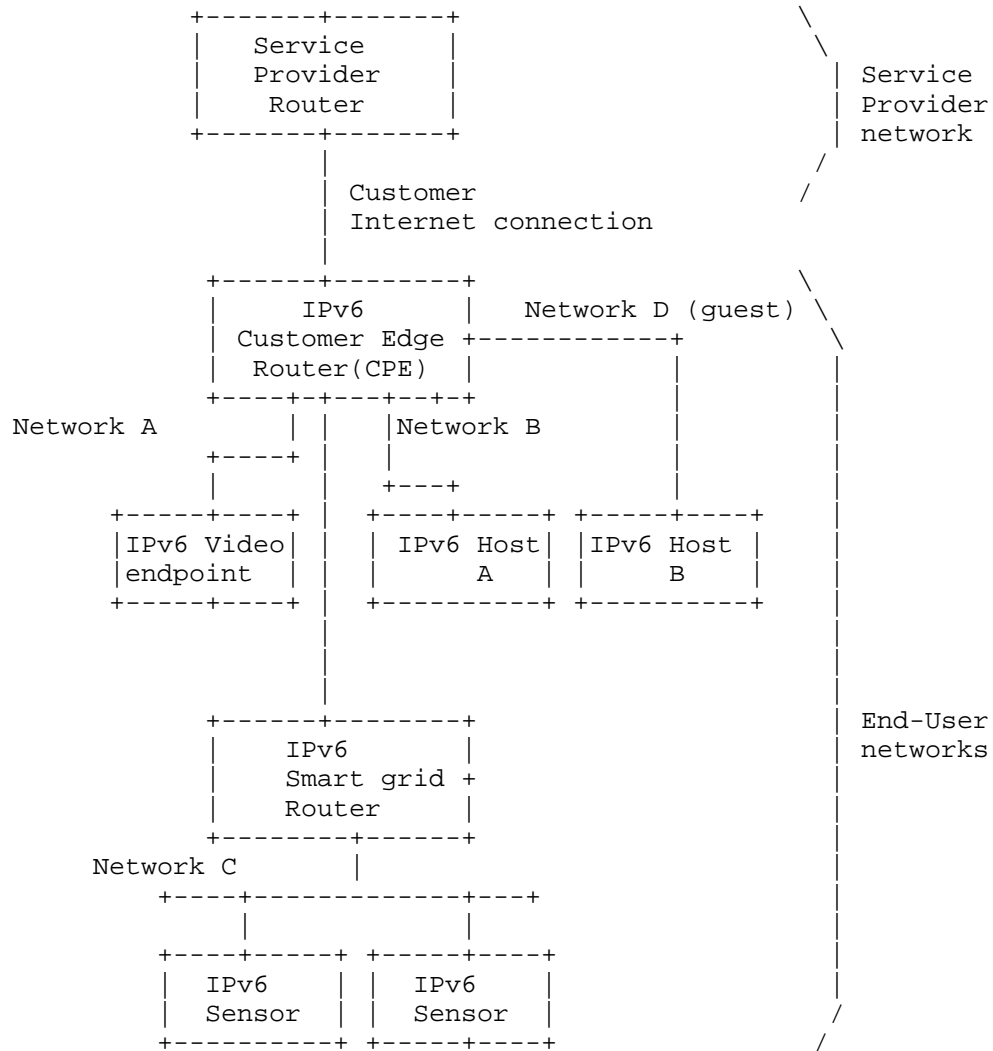


Figure 2

3.1. Class based prefix delegation

The Service Provider Router is preconfigured to provide prefixes from the following classes: "video", "default", "guest", and "smart-grid". It has a preconfigured policy to advertise prefixes to requesting routers based on the services supported by the service provider for a

given home. In the example home network, the CPE requests class based prefix allocation by sending a DHCPv6 SOLICIT message and include OPTION_PREFIX_CLASS in the OPTION_ORO.

The CPE receives an advertise with following prefixes in the IA_PD option :

1. IA_PD Prefix option with a prefix 3001::1::/64 containing OPTION_PREFIX_CLASS set to "video"
2. IA_PD Prefix option with a prefix 3001::2::/64 containing OPTION_PREFIX_CLASS set to "guest"
3. IA_PD Prefix option with a prefix 3001::3::/64 containing OPTION_PREFIX_CLASS set to "smart-grid"
4. IA_PD Prefix option with a prefix 3001::4::/64 containing OPTION_PREFIX_CLASS set to "default"

It sends a REQUEST message with all of above prefixes and receives a REPLY message.

3.2. IPv6 address assignment from class based prefix

The video endpoint in Network A in Figure 2 sends a DHCPv6 SOLICIT message requesting IA_NA address assignment with OPTION_USER_CLASS option containing the value "video" towards the CPE. The CPE assumes the role of the DHCPv6 server and sends an ADVERTISE to the video endpoint with OPTION_IA_NA containing an IPv6 address in OPTION_IAADDR from the "video" prefix class. The IPv6 address in the OPTION_IAADDR is set to 3001::1::1.

When the CPE receives a DHCPv6 SOLICIT requesting IA_NA for the IPv6 host from Network B, it offers an IPv6 address from the prefix class "default". For IPv6 host A it advertises 3001::4::1 as the IPv6 address in OPTION_IAADDR in response to the IA_NA request.

When the CPE receives a DHCPv6 SOLICIT requesting IA_NA for the IPv6 host from Network D (guest network), it offers an IPv6 address from the prefix class "guest". For IPv6 host B it advertises 3001::2::1 as the IPv6 address in OPTION_IAADDR in response to the IA_NA request. The Network D can be distinguished based on a preconfigured interface or SSID advertised by this CPE for guest hosts connecting to it.

3.3. IPv6 prefix delegation from class based prefix

The IPv6 Smart Grid router in Figure 2 sends a SOLICIT towards the CPE requesting prefix delegation in the "smart-grid" class by including the IA_PD option with the OPTION_PREFIX_CLASS containing "smart-grid". The CPE selects a longer prefix from "smart-grid" prefix previously obtained from Service Provider Router. It sends a DHCPv6 ADVERTISE message with IA_PD option containing the IPv6 prefix 3001:: 3:1::/96 and OPTION_PREFIX_CLASS set to "smart-grid". The Smart Grid router MAY then advertise that prefix in IPv6 Router Advertisement (RA) messages towards IPv6 sensors connected to it. IPv6 sensors can do SLAAC (as defined in [RFC4862]) to configure IPv6 address from the received RA message.

4. Acknowledgements

The authors would like to acknowledge review and guidance received from Frank Brockners, Wojciech Dec, Richard Johnson, Erik Nordmark, Hemant Singh, Mark Townsley, Ole Troan, Bernie Volz

5. IANA Considerations

IANA is requested to assign an option code to OPTION_PREFIX_CLASS from the "DHCPv6 and DHCPv6 options" registry (<http://www.iana.org/assignments/dhcpv6-parameters/dhcpv6-parameters.xml>).

6. Security Considerations

Security issues related to DHCPv6 which are described in section 23 of [RFC3315] and [RFC3633] apply for scenarios mentioned in this draft as well.

7. References

7.1. Normative References

- [I-D.baker-fun-multi-router]
Baker, F., "Exploring the multi-router SOHO network",
draft-baker-fun-multi-router-00 (work in progress),
July 2011.
- [I-D.baker-fun-routing-class]
Baker, F., "Routing a Traffic Class",
draft-baker-fun-routing-class-00 (work in progress),

July 2011.

- [I-D.chown-homenet-arch]
Arkko, J., Chown, T., Weil, J., and O. Troan, "Home Networking Architecture for IPv6", draft-chown-homenet-arch-00 (work in progress), September 2011.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.
- [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.
- [RFC3775] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

7.2. Informative References

- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.
- [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

Authors' Addresses

Shwetha Bhandari
Cisco Systems
Cessna Business Park, Sarjapura Marathalli Outer Ring Road
Bangalore, KARNATAKA 560 087
India

Phone:
Email: shwethab@cisco.com

Gaurav Halwasia
Cisco Systems
Cessna Business Park, Sarjapura Marathalli Outer Ring Road
Bangalore, KARNATAKA 560 087
India

Phone: +91 80 4426 1321
Email: ghalwasi@cisco.com

Sindhura Bandi
Cisco Systems
Cessna Business Park, Sarjapura Marathalli Outer Ring Road
Bangalore, KARNATAKA 560 087
India

Phone: +91 80 4426 2347
Email: sinb@cisco.com

Sri Gundavelli
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
USA

Email: sgundave@cisco.com

Hui Deng
China Mobile
53A, Xibianmennei Ave., Xuanwu District
Beijing 100053
China

Email: sinb@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 17, 2012

R. Droms
Cisco Systems
November 14, 2011

Modification to Default Value of MAX_SOL_RT
draft-droms-dhc-dhcpv6-maxsolrt-update-00

Abstract

This document updates RFC 3315 by redefining the default value for SOL_MAX_RT and defining an option through which a DHCPv6 server can override the client's default value for SOL_MAX_RT with a new value.

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, 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.

1. Introduction

Section 5.5 of the DHCPv6 specification [RFC3315] defines the default value of `MAX_SOL_RT` to be 120 seconds. In some circumstances, this default will lead to an unacceptably high volume of aggregated traffic at a DHCPv6 server.

1.1. 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].

2. Update to RC 3315

This document changes section 5.5 of RFC 3315 as follows:

OLD:

```
SOL_MAX_RT      120 secs  Max Solicit timeout value
```

NEW:

```
SOL_MAX_RT      3600 secs  Max Solicit timeout value
```

With this change, a DHCPv6 client that does not receive a satisfactory response will send Solicit messages with the same initial frequency and exponential backoff as specified in RFC 3315. However, the long term behavior of these DHCPv6 clients will be to send a Solicit message every 3600 seconds rather than every 120 seconds, significantly reducing the aggregated traffic at the DHCPv6 server.

The change to `MAX_SOL_RT` is in response to DHCPv6 message rates observed at a DHCPv6 server in a deployment in which many DHCPv6 clients are sending Solicit messages but the DHCPv6 server has been configured not to respond to those Solicit messages. RFC 3315 was written with the expectation that the 'M' and 'O' flags in NDP [RFC2461] would control the use of DHCPv6 by hosts. However, the current definition of the 'M' and 'O' flags in RFC 4861 [RFC4861] does not explicitly preclude the use of DHCPv6 by a host. Some devices are specified to initiate DHCPv6 even if RAs are received with the 'M' and 'O' bits set to 0. In some circumstances, it is desirable to enable the assignment of IPv6 addresses through DHCPv6 to some nodes on a link but not to others, which cannot be implemented through the 'M' and 'O' bits.

3. SOL_MAX_RT option

A DHCPv6 server sends the SOL_MAX_RT option to a client to override the default value of SOL_MAX_RT. One use for the SOL_MAX_RT option is to set a longer value for SOL_MAX_RT, which reduces the Solicit traffic from a client that has not received any IPv6 addresses.

The format of the SOL_MAX_RT option is:

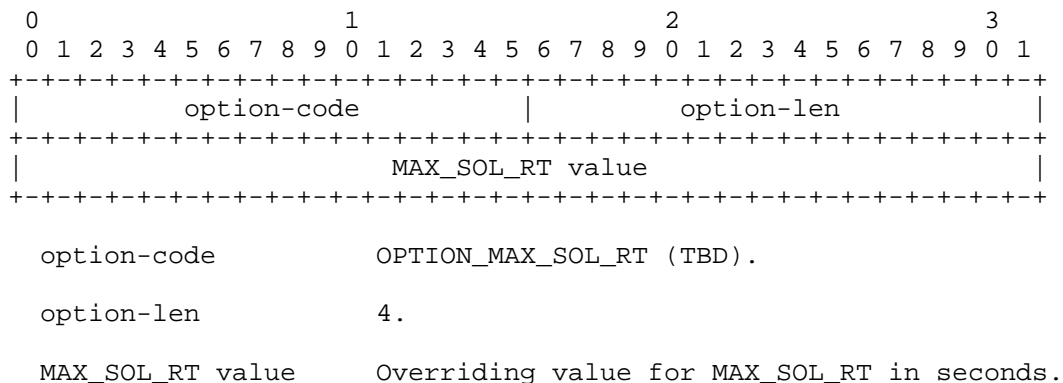


Figure 1

If the DHCPv6 server declines to assign any addresses to a client in an IA_NA or IA_TA option, it MAY include a SOL_MAX_RT option in the appropriate options field along with a Status Code option indicating NoAddrsAvail.

If a DHCPv6 client receives an IA_NA or IA_TA option containing a SOL_MAX_RT option, the client MUST set its internal SOL_MAX_RT parameter to the value contained in the SOL_MAX_RT option.

4. Security Considerations

This document introduces one security considerations beyond those described in RFC 3315. A malicious DHCPv6 server might cause a client to set its SOL_MAX_RT parameter to an arbitrarily high value with the SOL_MAX_RT option. Assuming the client also receives a response from a valid DHCPv6 server, the large value for SOL_MAX_RT will not have any effect.

5. IANA Considerations

IANA is requested to assign an options code from the "DHCP Option Codes" Registry for OPTION_MAX_SOL_RT.

6. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2461] Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.
- [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.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

Author's Address

Ralph Droms
Cisco Systems
1414 Massachusetts Avenue
Boxborough, MA 01719
USA

Phone: +1 978 936 1674
Email: rdroms@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 28, 2012

G. Halwasia
S. Bhandari
W. Dec
Cisco Systems
September 25, 2011

Client Hardware Address Option in DHCPv6
draft-halwasia-dhc-dhcpv6-hardware-addr-opt-00

Abstract

This document specifies the format and mechanism that is to be used for encoding client hardware address in DHCPv6 messages by defining a new DHCPv6 Client Hardware Address option.

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 March 28, 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	3
2. Problem Background and Scenario	3
3. DHCPv6 Client Hardware Address Option	4
4. DHCPv6 Client Behavior	4
5. DHCPv6 Relay Agent Behavior	4
6. DHCPv6 Server Behavior	5
7. IANA Considerations	5
8. Security Considerations	5
9. Acknowledgements	5
10. Normative References	5
Authors' Addresses	6

1. Introduction

This specification defines an optional mechanism and the related DHCPv6 option to allow DHCPv6 client or first hop DHCPv6 relay agent directly connected to the client to populate client hardware address in the DHCPv6 messages being sent towards the server.

2. Problem Background and Scenario

DHCPv4 protocol specification [RFC2131] provides a way to specify the client hardware address in the DHCPv4 message header. DHCPv4 message header has 'htype' and 'chaddr' fields to specify client hardware address type and hardware address respectively. The client hardware address thus learnt can be used by DHCPv4 server and relay in different ways. In some of the deployments DHCPv4 servers use 'chaddr' as a customer identifier and a key for lookup in the client lease database.

With the incremental deployment of IPv6 to existing IPv4 networks, effectievly an enablement of dual-stack, there will be devices that act as both DHCPv4 and DHCPv6 clients. In service provider deployments, a typical DHCPv4 implementation will use the client hardware address as one of the keys to build DHCP client lease database. In dual stack scenarios it is desirable for the operator to associate DHCPv4 and DHCPv6 messages as belonging to the same client interface based on an identifier that is already used by that operator such as the client hardware address.

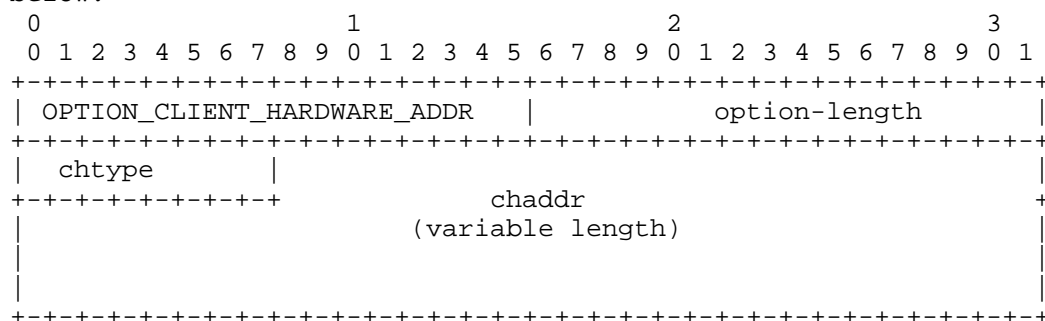
Currently, the DHCPv6 protocol specification [RFC3315] does not define a way for DHCP clients to specify client hardware address in the DHCPv6 message sent towards DHCPv6 Server. Similarly DHCPv6 Relay or Server cannot glean client hardware address from the contents of DHCPv6 message received. DHCPv6 protocol specification mandates all clients to prepare and send DUID as the client identifier option in all the DHCPv6 message exchange. However none of these methods provide a simple way to extract client's hardware address. This presents a problem to an operator who is using an existing DHCPv4 system with the client hardware address as the customer identifier, and desires to correlate DHCPv6 assignments using the same identifier. Modifying the system to use DUID based correlation across DHCPv4 and DHCPv6 is possible, but it requires a modification of the DHCPv4 system and associated back-ends.

Providing an option in DHCPv6 messages to carry client hardware address explicitly will help above mentioned scenarios. For e.g. it can be used along with other identifiers to associate DHCPv4 and DHCPv6 messages from a dual stack client. Further, having client

hardware address in DHCPv6 will help in proving additional information in event debugging and logging related to the client at relay and server.

3. DHCPv6 Client Hardware Address Option

The format of the DHCPv6 Client Hardware Address option is shown below.



option-code: OPTION_CLIENT_HARDWARE_ADDR (TBD)
option-length: 1 + length of chaddr
chtype: Client Hardware address type, see ARP section in "Assigned Numbers" RFC; e.g., '1' = 10mb ethernet.
chaddr: Client hardware address.

4. DHCPv6 Client Behavior

All hosts or clients MAY include DHCPv6 Client hardware address option in all the upstream DHCPv6 messages like SOLICIT, REQUEST, RENEW, REBIND, CONFIRM, RELEASE and DELCINE.

5. DHCPv6 Relay Agent Behavior

DHCPv6 Relay agents which are directly connected to clients/hosts MAY look for Client Hardware Address option in the incoming DHCPv6 client message. In absence of client hardware option, DHCPv6 Relay agents MAY include client hardware address option in relayed DHCPv6 (RELAY-FORW) message. The DHCPv6 Relay agent behaviour can depend on configuration that decides whether Client Hardware Address option needs to be processed and included.

In Relay chaining scenarios, any other relay agent other than first

hop DHCPv6 Relay agent or DHCPv6 LDRA [RFC6221] MUST not add this option.

6. DHCPv6 Server Behavior

If DHCPv6 Server is configured to store or use client hardware address, it MUST first look for the client hardware address option in the client DHCP message. In case it is not found, Server SHOULD look for client hardware option in the RELAY-FORW message of the DHCPv6 Relay agent closest to the client.

There is no requirement that a server return this option and its data in a downstream DHCP message.

7. IANA Considerations

IANA is requested to assign an option code to OPTION_CLIENT_HARDWARE_ADDR from the "DHCPv6 and DHCPv6 options" registry (<http://www.iana.org/assignments/dhcpv6-parameters/dhcpv6-parameters.xml>).

8. Security Considerations

Security issues related DHCPv6 are described in section 23 of [RFC3315].

9. Acknowledgements

10. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, 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.
- [RFC4361] Lemon, T. and B. Sommerfeld, "Node-specific Client Identifiers for Dynamic Host Configuration Protocol Version Four (DHCPv4)", RFC 4361, February 2006.

[RFC4580] Volz, B., "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Relay Agent Subscriber-ID Option", RFC 4580, June 2006.

[RFC6221] Miles, D., Ooghe, S., Dec, W., Krishnan, S., and A. Kavanagh, "Lightweight DHCPv6 Relay Agent", RFC 6221, May 2011.

Authors' Addresses

Gaurav Halwasia
Cisco Systems
Cessna Business Park, Sarjapura Marathalli Outer Ring Road
Bangalore, KARNATAKA 560 087
India

Phone: +91 80 4426 1321
Email: ghalwasi@cisco.com

Shwetha Bhandari
Cisco Systems
Cessna Business Park, Sarjapura Marathalli Outer Ring Road
Bangalore, KARNATAKA 560 087
India

Phone: +91 80 4426 0474
Email: shwethab@cisco.com

Wojciech Dec
Cisco Systems
Haarlerbergweg 13-19
1101 CH Amsterdam, Amsterdam 560 087
The Netherlands

Email: wdec@cisco.com

DHC Working Group
Internet-Draft
Updates: 2131 (if approved)
Intended status: Standards Track
Expires: February 17, 2012

N. Swamy
Nokia
G. Halwasia
P. Jhingran
Cisco Systems
August 16, 2011

Client Identifier Option in DHCP Server Replies
draft-ietf-dhc-client-id-01

Abstract

This document updates RFC2131 [RFC2131]. The changes to [RFC2131] defined in this draft clarifies the use of 'client identifier' option by the DHCP servers. The clarification addresses the issues arising out of the point specified by [RFC2131] that the server 'MUST NOT' return client identifier' option to the client.

Requirements

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].

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 February 17, 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	3
2. Problem Statement	3
3. Proposed Modification To [RFC2131]	4
4. IANA Considerations	4
5. Security Considerations	4
6. Acknowledgements	4
7. Normative References	5
Authors' Addresses	5

1. Introduction

The Dynamic Host Configuration Protocol (DHCP) defined in [RFC2131] provides configuration parameters to hosts on a TCP/IP based network. DHCP is built on a client-server model, where designated DHCP server allocate network addresses and deliver configuration parameters to dynamically configured hosts.

The changes to [RFC2131] defined in this document clarifies the use of 'client identifier' option by the DHCP servers. The clarification addresses the issues arising out of the point specified by [RFC2131] that the server 'MUST NOT' return client identifier' option to the client and thus facilitates DHCP relay agents and hosts to process downstream DHCP messages (DHCPOFFER, DHCPACK and DHCPNAK) when a DHCP client sets the 'chaddr' field as zero in DHCP request messages.

2. Problem Statement

[RFC2131] specifies that a combination of 'client identifier' or 'chaddr' and assigned network address constitute a unique identifier for the client's lease and are used by both the client and server to identify a lease referred in any DHCP messages. [RFC2131] also specifies that the server "MUST NOT" return 'client identifier' in DHCPOFFER and DHCPACK messages. DHCP relay agents and servers, following these recommendations MAY drop the DHCP packets in the absence of both 'client identifier' and 'chaddr'.

In some cases, client may not be having valid hardware address value to be filled in 'chaddr' field of the packet and hence may set this field as zero. One such example is when DHCP is used to assign IP address to a mobile phone or a tablet and where the 'chaddr' field is set to zero in DHCP request packets. In such cases, client usually sets the 'client identifier' option field (to a value as permitted in [RFC2131]), and both client and server use this field to uniquely identify the client with in a subnet.

Note that due to above mentioned recommendations in [RFC2131], valid downstream DHCP packets (DHCPOFFER, DHCPACK and DHCPNAK) from the server MAY get dropped at the DHCP relay agent in the absence of 'client identifier' option when 'chaddr' field is set as zero.

The problem may get aggravated when a client receives a response from the server without 'client identifier' and with 'chaddr' value set to zero, as it cannot guarantee that the response is intended for it. This is because even though the 'xid' field is present to map responses with requests, this field alone cannot guarantee that a particular response is for a particular client, as 'xid' values

generated by multiple clients within a subnet need not be unique.

This document attempts to address these problems faced by DHCP relay agent and client by proposing modification to DHCP server behavior. The proposed solution is in line with DHCPv6 [RFC3315] where the server always includes the Client Identifier option in the Reply messages.

3. Proposed Modification To [RFC2131]

If the 'client identifier' option is set in a message received from a client, the server MUST return the 'client identifier' option, unaltered, in its response message.

Following table is extracted from section 4.3.1 of [RFC2131] and relevant fields are modified accordingly to overcome the problems mentioned in this document.

Option	DHCPOFFER	DHCPACK	DHCPNAK
-----	-----	-----	-----
Client identifier (if sent by client)	MUST	MUST	MUST
Client identifier (if not sent by client)	MUST NOT	MUST NOT	MUST NOT

4. IANA Considerations

This memo asks the IANA for no new parameters.

5. Security Considerations

No known security considerations.

6. Acknowledgements

The authors would like to thank Bernie Volz, Ted Lemon, Barr Hibbs for their insightful discussions on the previous version of this document.

7. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, 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.

Authors' Addresses

Narasimha Swamy Nelakuditi
Nokia
Visiokatu 3
Tampere, 33720
Finland

Phone: +358 50487 2126
Email: narasimha.nelakuditi@nokia.com

Gaurav Halwasia
Cisco Systems
SEZ Unit, Cessna Business Park
Sarjapur Marathalli Outer Ring Road
Bangalore, 560103
India

Phone: +91 80 4426 1321
Email: ghalwasi@cisco.com

Prashant Jhingran
Cisco Systems
SEZ Unit, Cessna Business Park
Sarjapur Marathalli Outer Ring Road
Bangalore, 560103
India

Phone: +91 80 4426 1800
Email: pjhingra@cisco.com

dhc
Internet-Draft
Intended status: Standards Track
Expires: January 12, 2012

T. Lemon
Nominum, Inc.
H. Deng
L. Huang
China Mobile
July 11, 2011

Relay Agent Encapsulation for DHCPv4
draft-ietf-dhc-dhcpv4-relay-encapsulation-01

Abstract

This document describes a general mechanism whereby DHCP relay agents can encapsulate DHCP packets that they are forwarding in the direction of DHCP servers, and decapsulate packets that they are forwarding toward DHCP clients, so that more than one relay agent can insert relay agent suboptions into the forwarding chain.

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 12, 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
1.1.	Requirements Language	4
1.2.	Terminology	4
2.	Protocol Summary	6
2.1.	RELAYFORWARD Message	6
2.2.	RELAYREPLY Message	6
2.3.	Layer Two Address suboption	6
3.	Encoding	7
3.1.	The fixed-length header	8
3.2.	Relay Segment	9
3.3.	Encapsulation Segment	9
4.	DHCP Relay Agent Behavior	9
4.1.	Packet processing	10
4.1.1.	Packets traveling toward DHCP servers	11
4.1.2.	Packets traveling toward DHCP clients	11
4.1.3.	Anti-spoofing	11
4.2.	Constructing RELAYFORWARD messages	11
4.2.1.	Initializing the fixed-length header	11
4.2.2.	Initializing the relay segment	12
4.2.3.	Fixed header settings for RELAYFORWARD messages	12
4.2.4.	Fixed header settings for BOOTREQUEST messages	13
4.2.5.	Initializing the encapsulation segment	13
4.3.	Decapsulating RELAYREPLY messages	13
4.3.1.	Processing relay agent suboptions	13
4.3.2.	Constructing the decapsulated message	14
4.4.	Retransmitting modified messages	14
4.4.1.	Layer two relay agents	14
4.4.1.1.	Constructing the headers	14
4.4.1.2.	Forwarding the modified packet	15
4.4.2.	Layer three relay agents	15
4.4.2.1.	Transmitting a decapsulated RELAYREPLY message	15
4.4.2.2.	Transmitting a decapsulated BOOTREPLY message	16
4.4.2.3.	Transmitting other messages	16
5.	DHCP Server Behavior	16
5.1.	Receiving RELAYFORWARD messages	16
5.1.1.	Decapsulation	16
5.1.2.	Processing of decapsulated suboptions	16
5.1.3.	Address allocation	17
5.1.3.1.	Default link selection algorithm	17
5.1.3.2.	Other link selection algorithms	18
5.2.	Responding to RELAYFORWARD messages	18
5.2.1.	Constructing a RELAYREPLY encapsulation	18

- 5.2.1.1. Constructing the relay segments 19
- 5.2.1.2. Constructing the fixed-length header 19
- 5.2.2. Transmission of RELAYREPLY messages 19
- 5.3. Responding to messages other than RELAYFORWARD 20
- 6. DHCP Client Behavior 20
- 7. Security Considerations 20
- 8. IANA Considerations 21
- 9. References 21
 - 9.1. Normative References 21
 - 9.2. Informative References 22
- Authors' Addresses 22

1. Introduction

In some networking environments, it is useful to be able to configure relay agents in a hierarchy, so that information from a relay agent close to the client can be combined with information from one or more relay agents closer to the server, particularly in cases where the relay agents may be in separate administrative domains.

The current Relay Agent Information Option (RAIO) specification [RFC3046] specifies that when a relay agent receives a packet containing an RAIO, it must not add an RAIO. This prevents chaining of RAIOS, and hence prohibits the hierarchical use case.

DHCP version 6 [RFC3315] provides a much cleaner technique for providing RAIO suboptions to the DHCP server. Rather than appending an information option to the DHCP client's message, the relay agent encapsulates the DHCP client message in a new DHCP message which it sends to the DHCP server along with any options it chooses to add to provide information to the DHCP server.

This document specifies a mechanism for providing the same functionality in DHCPv4.

1.1. 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].

1.2. Terminology

The following terms and acronyms are used in this document:

BOOTREPLY message	Any DHCP or BOOTP message in which the 'op' field is set to BOOTREPLY.
BOOTREQUEST message	Any DHCP or BOOTP message in which the 'op' field is set to BOOTREQUEST.
DHCP	Dynamic Host Configuration Protocol Version 4 [RFC2131]
decapsulate	the act of de-encapsulating DHCP packets being relayed from DHCP servers or relay agents in the direction of DHCP clients, according to this specification.

encapsulate	the act of encapsulating DHCP packets that are being relayed from DHCP clients or relay agents toward DHCP servers, according to the method described in this specification.
encapsulating relay agent	a relay agent that implements this specification and is configured to encapsulate.
L2RA	Layer 2 Relay Agent--a relay agent that doesn't have an IP address reachable by the DHCP server.
L3RA	Layer 3 Relay Agent--a relay agent that has an IP address reachable by the DHCP server.
option buffer	the portion of the DHCP packet following the magic cookie in the 'vend' field of the DHCP Packet.
RAIO	Relay Agent Information Option [RFC3046]. Also commonly referred to as "Option 82."
RAIO suboption	a DHCP suboption that has been defined for encapsulation in the Relay Agent Information Option
relay message	a RELAYFORWARD or RELAYREPLY message.
RELAYFORWARD message	Any DHCP or BOOTP message in which the 'op' field is set to RELAYFORWARD.
RELAYREPLY message	Any DHCP or BOOTP message in which the 'op' field is set to RELAYREPLY.
silently discard	in many places in this specification, the implementation is required to silently discard erroneous messages. What is meant by 'silently discard' is that the implementation MUST NOT send any ICMP message indicating that the delivery was in error. It may be desirable for the implementation to keep a count of messages that have been discarded, either by message type or by reason for discarding, or some combination. Nothing in this specification should be construed to forbid such data collection.

2. Protocol Summary

This document specifies two new BOOTP message types: the RELAYFORWARD message, and the RELAYREPLY message. These messages are analogous to the Relay Forward and Relay Reply messages in DHCPv6 [RFC3315].

Although this specification is generally aimed at DHCP implementations, it is not specifically restricted to DHCP, and is applicable to BOOTP in cases where the BOOTP server is a DHCP server that implements this specification, or the less likely case that the BOOTP server only supports the BOOTP protocol, but still implements this specification.

In general, when the term "DHCP" appears in this specification, the reader should not read this as intending to exclude BOOTP.

2.1. RELAYFORWARD Message

Conforming relay agents encapsulate messages being sent toward DHCP servers in RELAYFORWARD messages.

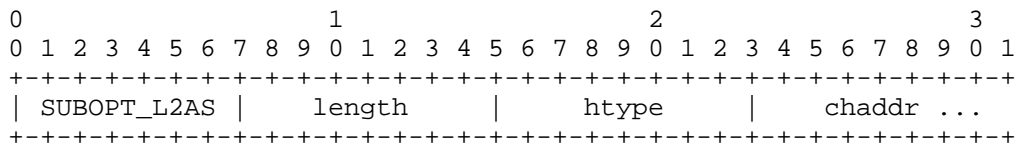
2.2. RELAYREPLY Message

A conforming DHCP server encapsulates any message being sent toward a DHCP client in a RELAYREPLY message, if the message being responded to was encapsulated.

A conforming relay agent, when it receives a RELAYREPLY message, decapsulates the message contained in the RELAYREPLY message and sends it to the next relay or to the client.

2.3. Layer Two Address suboption

In cases where the closest relay agent to the client is an L2RA, but where there is an L3RA on the path to the client, the DHCP server will encode the link layer address that would normally go in the chaddr field of the DHCP packet into a Layer Two Address suboption.



The Layer Two Address suboption has four fields:

SUBOPT_L2AS One octet: the suboption code for the Layer Two Address suboption (TBD).

length One octet: the length of the Layer Two Address suboption.

htype One octet: the type of the Layer Two Address suboption-- corresponds to the 'htype' field in a non-relay DHCP or BOOTP message.

chaddr One or more octets: the layer two address of the client, from the 'chaddr' field of the DHCP or BOOTP message.

3. Encoding

RELAYFORWARD and RELAYREPLY messages are distinguished through the use of the 'op' field of the DHCP packet.

In non-relay DHCP packets, the 'op' field either contains BOOTREQUEST, for any DHCP message from the client to the server, or BOOTREPLY, for any DHCP message from the server to the client.

This document defines two additional codes, RELAYFORWARD and RELAYREPLY. Conforming DHCP servers and DHCP relay agents MUST support these two new values for the 'op' field. DHCP clients should never see either value.

code	meaning
1	BOOTREQUEST
2	BOOTREPLY
3	RELAYFORWARD
4	RELAYREPLY

Values for the 'op' field

RELAYFORWARD and RELAYREPLY messages share only the 'op' field in common with other DHCP and BOOTP messages. The remainder of the message consists of a series of fixed-length fields followed by two variable-length fields: the relay segment, and the encapsulated message.

```

+-----+-----+-----+-----+
|  op  |  ep  |  padlen  |
+-----+-----+-----+-----+
|  rslen  |  caplen  |
+-----+-----+-----+-----+
|                aiaddr                |
+-----+-----+-----+-----+
.
.   relay segment   .
.
+-----+-----+-----+-----+
.
.   encapsulated message   .
.
+-----+-----+-----+-----+

```

3.1. The fixed-length header

The fixed-length header of the relay message contains a series of fields that perform two purposes: to provide enough information that the DHCP server can reconstruct the original packet sent by the DHCP client, and to establish the lengths of the two variable-length segments.

To satisfy the first of these requirements, two fields in the fixed-length header report the amount of padding stripped from the client message, if any, and whether or not an end option was stripped from the client message. Except for a relay message that immediately encapsulates a message from a DHCP client, these fields are always zero. Using these two fields, the DHCP server can reconstruct the client packet exactly, and this allows the DHCP server to validate any signature [RFC3118] that may be present.

The fixed-length header consists of five fields:

op The BOOTP 'op' field, which, for a relay message, MUST contain the RELAYFORWARD or RELAYREPLY code.

ep If an End option was present in the option buffer prior to encapsulation, this field is set to 1; otherwise, it is set to 0. This field is a single byte.

padlen The length of any padding that was removed from the option buffer prior to encapsulation: two bytes in network byte order.

rslen The length of the relay segment: two byte in network byte order.

caplen The length of the encapsulation segment: two byte in network byte order.

aiaddr Relay agent IP address.

3.2. Relay Segment

The relay segment contains any RAIIO suboptions that the encapsulating agent (the relay agent or the DHCP server) wishes to send. End and Pad options MUST NOT appear in the relay segment.

3.3. Encapsulation Segment

The encapsulation segment contains the entire DHCP message being encapsulated, with four exceptions:

- o The encapsulating agent MUST omit the IP and UDP headers, as well as any layer two header, from the encapsulated message.
- o The encapsulating agent MUST omit any options following the first End option in the option buffer. These options are assumed to be garbage, and are not covered by any signature [RFC3118].
- o The encapsulating agent MUST omit any Pad options present either at the end of the option buffer, or prior to the first End option, that are followed only by other Pad options or a single End option. The encapsulating agent MUST record number of Pad options that were omitted in the 'padlen' field of the message header.
- o The encapsulating agent MUST omit the End option, if present. The encapsulating agent MUST set the 'ep' field in the message header to 1 if an End option was present in the option buffer, and to zero if no End option was present.

These exceptions apply only to the option buffer. The encapsulating agent MUST NOT modify the contents of the 'file' and 'sname' fields. The encapsulating agent MUST NOT count End or Pad options that appear in these fields.

4. DHCP Relay Agent Behavior

DHCP Relay agents implementing this specification MUST have a configuration parameter controlling relay encapsulation. By default, relay encapsulation MUST be disabled.

Relay agents with encapsulation disabled MUST NOT encapsulate. Relay agents with encapsulation disabled MUST NOT decapsulate.

In any case where a relay agent implementing this specification does not encapsulate or decapsulate, it MUST behave exactly as a relay agent that does not implement this specification at all.

DHCP relay agents that are configured with encapsulation enabled, but which have no agent-specific options to send to the DHCP server, MUST encapsulate. Relay agents that are configured with encapsulation enabled MUST decapsulate.

Layer two relay agents MUST silently discard any messages that contains an IPsec authentication header [RFC4302]. This is because they cannot modify such messages, but also cannot detect that a message from the DHCP server is in response such messages, since the response message might not contain an IPsec authentication header.

If a relay message would exceed the MTU of the outgoing interface, it MUST be discarded, and an error condition SHOULD be logged.

4.1. Packet processing

Relay agents implementing this specification may receive packets directed toward DHCP servers with a source port of 67 (BOOTPS). Therefore, the source port cannot be used to determine whether the packet is traveling toward a DHCP server or toward a DHCP client.

In order to determine whether a message is traveling toward a DHCP client or toward a DHCP server, the relay agent must check the 'op' field of the DHCP message. If the 'op' field is set to BOOTREQUEST or RELAYFORWARD, the message is traveling toward a DHCP server. If the 'op' field is set to BOOTREPLY or RELAYREPLY, the message is traveling toward a DHCP client. At the time of the writing of this specification, no other value is meaningful in the 'op' field.

Relay agents implementing this specification MUST NOT encapsulate or decapsulate messages with other values in the 'op' field. It is assumed that if meanings are defined for additional values, the document that specifies the meaning of those values will update this document; in the absence of such an update, the behavior specified here will remain in effect.

Relay agents implementing this specification MAY differentiate between DHCP and BOOTP messages. Under normal circumstances, BOOTP and DHCP messages are forwarded to the same server, which should be able to successfully decapsulate both DHCP and BOOTP messages. However, some relay agents may send BOOTP and DHCP packets to

different servers; this document should not be construed to require that such a relay agent should encapsulate all messages regardless of protocol.

4.1.1. Packets traveling toward DHCP servers

Any DHCP or BOOTP packet with an 'op' value of BOOTREQUEST or RELAYFORWARD is traveling toward a DHCP server. When a DHCP relay agent that is configured to encapsulate receives such a packet, the relay agent MUST encapsulate that packet into a RELAYFORWARD message and send it to the address or addresses with which it is configured to forward messages intended for DHCP servers.

4.1.2. Packets traveling toward DHCP clients

Any DHCP or BOOTP packet with an 'op' value of BOOTREPLY or RELAYREPLY is traveling toward a DHCP client. When a DHCP relay agent that is configured to encapsulate receives a RELAYREPLY message that is traveling toward a DHCP or BOOTP client, the relay agent MUST decapsulate that message and forward the decapsulated message toward the client.

4.1.3. Anti-spoofing

Because this specification allows for chaining of relay agent-supplied information, it is now possible for a relay agent outside of the trusted portion of a network to communicate relay agent information to the DHCP server without preventing the legitimate relay from communicating return path information to the DHCP server, as is the case with RFC3046.

In order to prevent this sort of spoofing, relay agents implementing this specification MUST be configurable to discard all RELAYFORWARD messages that they receive. Administrators relying on a trusted network architecture to control the flow of information to the DHCP server SHOULD configure relay agents on the edge of their networks to discard RELAYFORWARD messages.

4.2. Constructing RELAYFORWARD messages

4.2.1. Initializing the fixed-length header

The relay agent constructs the RELAYFORWARD message by constructing the fixed-length header as specified in the earlier section titled 'Encoding'. The relay agent MUST set the 'op' field to a value of RELAYFORWARD.

If the relay agent is not a layer two relay agent

[I-D.ietf-dhc-l2ra], it MUST store one of its own IP addresses in the 'aiaddr' field. This address MUST be a valid IP address that is reachable by the next hop relay(s) or DHCP server(s) to which the relay agent is configured to forward.

DHCP servers normally use the relay agent IP address to determine on what link the DHCP client's IP address should be allocated. In some cases, the value stored in the 'aiaddr' field will not be a valid IP address on the link on which the source message was received. In this case, the relay agent MUST include a link selection suboption [RFC3527] that identifies that link in the relay segment.

If the relay agent is a layer two relay agent, it MAY include a link selection suboption in the relay segment.

If the message being encapsulated is a BOOTREQUEST, L2RAs MUST store a value of zero in the 'aiaddr' field. Otherwise, the L2RA MUST copy the value of the 'aiaddr' field in the RELAYFORWARD message being encapsulated into the 'aiaddr' field of the RELAYFORWARD message that it generates.

The 'rslen' field depends on the length of the relay segment. The 'caplen', 'padlen' and 'ep' values in the fixed header are initialized differently depending on whether the message being encapsulated is a BOOTREQUEST or a RELAYFORWARD message.

4.2.2. Initializing the relay segment

Following the fixed header, the relay agent MUST append any RAI0 suboptions it wishes to send to the DHCP server; this is the 'relay segment'. It MUST store the length of the relay segment in the 'rslen' field of the fixed header.

The relay agent SHOULD include a Relay Agent ID suboption [I-D.ietf-dhc-relay-id-suboption] in the relay segment to identify itself to the DHCP server.

4.2.3. Fixed header settings for RELAYFORWARD messages

If the message being encapsulated is a RELAYFORWARD message, the relay agent MUST initialize the 'caplen' field of the fixed header to the length of the source message, excluding any layer 2, IP and UDP headers. It MUST append the contents of the message, again excluding any layer 2, IP or UDP headers, to the new message. It MUST initialize the 'ep' and 'padlen' fields in the fixed header of the new message to zero.

4.2.4. Fixed header settings for BOOTREQUEST messages

If the message being encapsulated is a BOOTREQUEST message, the relay agent determines the length of the encapsulation segment by scanning forward across the option buffer of the source message, beginning with the first option in the option buffer, until an End option is reached, or the end of the buffer is reached. The difference between the offset of this location in the message and the offset of the first location following the UDP header of the message is the length of the message to be relayed.

If an End option terminated the scan, the relay agent MUST set the value of the 'ep' field in the fixed header to one. Otherwise, the relay agent MUST set the value of the 'ep' field to zero.

The relay agent MUST count all of the Pad options that follow the last option in the option buffer that is neither a Pad nor an End option. The relay agent MUST store this count in the 'padlen' field of the fixed header.

The relay agent MUST store the difference between the value stored in 'padlen' and the length of the message to be relayed in the 'caplen' field of the fixed header.

4.2.5. Initializing the encapsulation segment

The relay agent MUST copy the portion of the message being encapsulated that immediately follows the UDP header into the RELAYFORWARD message being generated. The length of the data being copied is the value that was stored in 'caplen'.

4.3. Decapsulating RELAYREPLY messages

To decapsulate a RELAYREPLY message, the relay agent creates a decapsulated message, processes any RAIIO suboptions it recognizes in the relay segment, and forwards the decapsulated message to its destination.

4.3.1. Processing relay agent suboptions

The suboptions parsed from the relay segment are processed by the relay agent as specified in the Relay Agent Information Option specification [RFC3046], as well as in any documents that define suboptions to the Relay Agent Information Option. A current list of DHCP Relay Agent suboptions and the documents that define them can be located in the IANA protocol registry for Bootp and DHCP parameters, in the section titled "DHCP Relay Agent Sub-Option Codes."

4.3.2. Constructing the decapsulated message

To construct a decapsulated message, the relay agent copies the portion of the RELAYREPLY message following the relay segment, with a length specified in the 'caplen' field of the fixed-length header, into the new message.

4.4. Retransmitting modified messages

If the relay agent did not modify the message either by encapsulating or decapsulating it, it retransmits the message according to existing practice.

Otherwise, how the modified message is transmitted to its next destination depends on two factors. First, is the relay agent that modified the message a layer two [I-D.ietf-dhc-l2ra] relay agent or a layer three [RFC1542] relay agent? Second, is the modified message a BOOTREPLY message, a RELAYREPLY message, or some other message?

4.4.1. Layer two relay agents

There are two special aspects to the handling of relayed packets in Layer Two Relay Agents (L2RAs). The first is the construction of the layer two, IP and UDP headers on the packet. The second is how the L2RA makes the forwarding decision.

4.4.1.1. Constructing the headers

The L2RA constructs the headers on the relayed packet by copying and, as necessary, modifying the headers from the original packet.

If there is a layer two header, the L2RA MUST copy this header in accordance with the needs of the particular layer two implementation it is using. If the header contains a packet length field, the L2RA MUST adjust the value in the packet length field. If the header contains a non-secure integrity check such as a CRC or checksum that covers the entire packet, the L2RA MUST recompute this value.

L2RA encapsulation in cases where the layer two contains a secure integrity check must either construct a new integrity signature, or else remove the integrity signature. If neither of these is possible, the L2RA MUST silently discard the packet.

The L2RA MUST copy the IP header without modification except length and checksum field which should be recomputed. If the IP header contains any sort of secure integrity check on the packet, the L2RA MUST silently discard the packet.

The L2RA MUST copy the UDP header and adjust the 'Length' field [RFC0768]. If the contents of the 'Checksum' field are not zero, the L2RA MUST compute a new checksum according to the algorithm specified in User Datagram Protocol. [RFC0768]

4.4.1.2. Forwarding the modified packet

Ordinarily when a layer two device forwards a packet, it simply copies that packet from the interface on which it was received and transmits it, unchanged, on one or more interfaces other than that interface. The mechanism used to choose which interfaces it forwards the packet to is beyond the scope of this document.

Once a DHCP packet has been modified by the L2RA either as an encapsulation or a decapsulation, the L2RA must forward it either toward the DHCP server or the DHCP client. The implementation has two options: transmit the packet as it would transmit any other packet, or use its configuration and/or information in the relay header to direct the packet out a particular interface.

The details of ordinary packet forwarding in devices that implement L2RA is beyond the scope of this document.

When processing a RELAYREPLY message, the L2RA MAY use information in the relay segment of the RELAYREPLY to determine on which network interface the RELAYREPLY should be forwarded.

When processing any other message, the L2RA MAY use configuration information to direct the packet out a specific port or ports that have been marked as reaching DHCP servers. The L2RA MUST NOT forward any packet on the interface on which it was received, even if that interface is so marked.

4.4.2. Layer three relay agents

4.4.2.1. Transmitting a decapsulated RELAYREPLY message

When the decapsulated message is itself a RELAYREPLY message, the relay agent MUST forward the decapsulated message to the IP address specified in the 'aiaddr' field of the fixed-length header.

If the relay segment of the packet that was decapsulated contains a Link Layer Address suboption, the relay agent MUST transmit the packet to that link layer address without attempting to use Address Resolution Protocol (ARP) to translate the address contained in 'aiaddr' to a layer two address.

4.4.2.2. Transmitting a decapsulated BOOTREPLY message

When transmitting a decapsulated BOOTREPLY message, the relay agent transmits the message as specified in Bootstrap Protocol, Section 4 [RFC0951].

4.4.2.3. Transmitting other messages

When transmitting RELAYFORWARD and BOOTREQUEST messages, the relay agent simply sends the message to the IP address or addresses configured as DHCP servers for that relay agent.

5. DHCP Server Behavior

A DHCP server which receives a RELAYREPLY message MUST silently discard that message.

5.1. Receiving RELAYFORWARD messages

DHCP servers that implement this specification MUST decapsulate RELAYFORWARD messages.

5.1.1. Decapsulation

By design, this specification supports multiple layers of encapsulation. The DHCP server implementation therefore MUST decapsulate each layer and retain the information in that layer, organized so that the ordering of the encapsulation is available both during packet processing, and when constructing the reply.

Aside from the necessity of handling an RAI0 differently than an encapsulation when constructing a RELAYREPLY message, DHCP servers MUST NOT, by default, treat relay suboptions received in an RAI0 any differently than relay suboptions received in an encapsulation.

It is not the goal of this specification to define a particular implementation strategy or data structure for representing the encapsulation, so long as the ability to process the options and suboptions as required below is preserved. Implementations MAY provide means for addressing the contents of relay segments and of the inner encapsulation segment in any convenient form, as long as it conforms generally to the requirements of this document.

5.1.2. Processing of decapsulated suboptions

This section specifies requirements for the processing of decapsulated relay suboptions in the default case, where no custom

processing has been specified. This should not be construed to forbid implementations for providing mechanisms for customizing the processing of these suboptions.

This document does not specify special handling for DHCP options. Only the inner encapsulation--the encapsulation of the original message sent from the client, which is done by the first encapsulating relay--can ever contain DHCP Options; hence, whatever normal mechanisms a DHCP server provides for processing these options should suffice.

Some relay agent suboptions, such as the Relay Agent Subnet Selection suboption [RFC3527], are intended to be processed by DHCP servers. Others, like the Circuit ID and Remote ID [RFC3046] suboptions, were not intended to be processed by the DHCP server, but are often used by the DHCP server for identification purposes.

In cases where more than one relay agent sends the same suboption, the DHCP server must either factor all such suboptions into its processing, or choose one such suboption and use that exclusively for processing.

By default, DHCP servers MUST use the outermost suboption (the one added by the relay agent closest to the DHCP server) for every suboption that was sent by more than one relay agent.

5.1.3. Address allocation

During normal processing, DHCP servers use a variety of data to determine where the DHCP client is in the network topology. These data are provided by relay agents. In the absence of any relay-agent-provided topology data, the DHCP server assumes that the client is connected to the link on which the message was received.

One datum used by DHCP servers in locating the client is the value of the 'giaddr' field of the BOOTP header. This specification eliminates the use of giaddr; hence, it cannot be used in the address allocation decision.

The functionality provided by giaddr is replaced in this specification by the 'aiaddr' field in the fixed-length relay header.

5.1.3.1. Default link selection algorithm

DHCP servers MUST implement a default algorithm for determining the link to which the client is attached. This algorithm is to first search the client message for a subnet selection option [RFC3011].

The server next searches for link-identifying data in each RELAYFORWARD encapsulation, starting from the inner most and ending at the outermost, until a RELAYFORWARD is found that identifies the client's location.

A RELAYFORWARD encapsulation contains link-identifying data if the value of the 'aiaddr' field of the fixed-length header is not zero, or if the relay segment contains a Link Selection suboption [RFC3527].

If a Link Selection suboption is present in the innermost RELAYFORWARD message containing link-identifying data, the DHCP server MUST use the contents of that option to identify the link to which the client is connected.

Otherwise, if a Subnet Selection option was found in the client message, the DHCP server MUST use the contents of that option to identify the link to which the client is connected.

Otherwise, the DHCP server MUST use the contents of the 'aiaddr' field in the RELAYFORWARD encapsulation that was identified as being the innermost RELAYFORWARD containing link-identifying data.

5.1.3.2. Other link selection algorithms

DHCP servers implementing this specification MAY implement link selection algorithms other than the one described in the preceding section. DHCP servers MUST NOT use any link selection algorithm other than the one described in the preceding section unless specially configured to do so.

5.2. Responding to RELAYFORWARD messages

Once the DHCP server has processed the encapsulated message from the DHCP client and constructed a response to the DHCP client, it constructs a RELAYREPLY message and sends it toward the client.

5.2.1. Constructing a RELAYREPLY encapsulation

The server MUST encapsulate any response to a client message contained in one or more RELAYFORWARD encapsulations in a set of corresponding RELAYREPLY encapsulations. Each RELAYREPLY is nested in the same way that the corresponding RELAYFORWARD was nested, so that the innermost RELAYREPLY corresponds to the innermost RELAYFORWARD, and the outermost RELAYREPLY corresponds to the outermost RELAYFORWARD.

5.2.1.1. Constructing the relay segments

The server MUST copy every suboption that appeared in the relay segment of the RELAYFORWARD encapsulation into corresponding outgoing RELAYREPLY encapsulation's relay segment. The server SHOULD NOT preserve the order of options from the incoming relay segment to the outgoing relay segment.

If the message encapsulated by a particular RELAYREPLY encapsulation is not a RELAYREPLY, or if the message encapsulated by that RELAYREPLY is a RELAYREPLY, but the 'aiaddr' field of the fixed-length header of the inner RELAYREPLY has a value of zero, the DHCP server MUST include a Layer Two Address suboption in the relay segment. The DHCP server MUST set the 'htype' field of the Layer Two Address suboption to the value of 'htype' in the client message. The DHCP server MUST set the 'chaddr' field in the Layer Two Address suboption to the value of the 'chaddr' field in the client message.

5.2.1.2. Constructing the fixed-length header

The server MUST set the values of 'ep' and 'padlen' in the fixed-length header to zero. The server MUST set the value of 'caplen' to the length of the message encapsulated in the RELAYREPLY encapsulation being constructed. The server MUST set the value of 'rslen' to the length of the relay segment of the RELAYREPLY encapsulation being constructed.

If the message encapsulated in the RELAYREPLY being constructed is a RELAYREPLY, and the 'aiaddr' field of the RELAYFORWARD encapsulation corresponding to the inner RELAYREPLY is not zero, the DHCP server MUST copy the value from that 'aiaddr' field to the 'aiaddr' field of the RELAYREPLY being constructed.

Otherwise, if the BROADCAST bit in the 'flags' field of the inner message is set to 1, or 'ciaddr' is zero and 'yiaddr' is also zero, the DHCP server MUST set the value of 'aiaddr' to 255.255.255.255. If 'ciaddr' is not zero, the DHCP server must copy that value to 'aiaddr'. If 'ciaddr' is zero and 'yiaddr' is not, the DHCP server MUST copy the value of 'yiaddr' to 'aiaddr'.

5.2.2. Transmission of RELAYREPLY messages

If the value of 'aiaddr' in the outermost RELAYFORWARD encapsulation to which the RELAYREPLY is a response is nonzero, the DHCP server MUST transmit the RELAYREPLY to that IP address.

Otherwise, if 'ciaddr' and 'yiaddr' are both zero, or the BROADCAST bit in the 'flags' field is set to 1, or the DHCP server

implementation is not able to perform the ARP-cache preloading trick described in Bootstrap Protocol, Section 4 [RFC0951], the DHCP server MUST broadcast the RELAYREPLY message to the 255.255.255.255 broadcast address.

Otherwise, the DHCP server MUST use the value of 'ciaddr', if nonzero, or 'yiaddr', otherwise, as the destination address for the message, and MUST preload its ARP cache (or otherwise arrange to transmit the message without using ARP) to the layer two address provided by the client in 'htype' and 'chaddr' and 'hlen'.

5.3. Responding to messages other than RELAYFORWARD

When a DHCP server constructs a response to a DHCP client message that did not arrive encapsulated in a RELAYFORWARD message, the DHCP server MUST NOT encapsulate the response in a RELAYREPLY message. DHCP server implementors should be careful that their servers do not respond to an incoming packet with RAIIO from a non-conforming relay agent with a RELAYREPLY message.

6. DHCP Client Behavior

A DHCP client that receives either a RELAYFORWARD message or a RELAYREPLY message MUST silently discard that message.

7. Security Considerations

DHCP Relay Information Option [RFC3046] limits relay agent information to a single relay agent, and provides some minimal anti-spoofing. By supporting relay agent chaining, it is now possible for a relay agent downstream of the trusted portion of a provider network to communicate relay agent information options to a DHCP server.

This specification defines a much more robust spoofing rejection mechanism, by allowing the administrator to configure the relay to discard RELAYFORWARD messages originating from outside of the trusted portion of the network. Administrators of networks that rely on this trusted/untrusted configuration are encouraged to configure edge relays to discard RELAYFORWARD messages.

In networks where trusted relay agents may be separated from the DHCP server by transit networks that are not trusted, it is possible to modify the message in transit. This possibility exists with normal DHCP relays as well. Administrators of such networks should consider using relay agent authentication [RFC4030] to prevent modification of relay agent communications in transit.

8. IANA Considerations

We request that IANA assign one new suboption code from the registry of DHCP Agent Sub-Option Codes maintained in <http://www.iana.org/assignments/bootp-dhcp-parameters>. This suboption code will be assigned to the Layer Two Address Suboption.

9. References

9.1. Normative References

- [I-D.ietf-dhc-relay-id-suboption] Stapp, M., "The DHCPv4 Relay Agent Identifier Suboption", draft-ietf-dhc-relay-id-suboption-07 (work in progress), July 2009.
- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.
- [RFC0951] Croft, B. and J. Gilmore, "Bootstrap Protocol", RFC 951, September 1985.
- [RFC1542] Wimer, W., "Clarifications and Extensions for the Bootstrap Protocol", RFC 1542, October 1993.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.
- [RFC3011] Waters, G., "The IPv4 Subnet Selection Option for DHCP", RFC 3011, November 2000.
- [RFC3046] Patrick, M., "DHCP Relay Agent Information Option", RFC 3046, January 2001.
- [RFC3118] Droms, R. and W. Arbaugh, "Authentication for DHCP Messages", RFC 3118, June 2001.
- [RFC3527] Kinnear, K., Stapp, M., Johnson, R., and J. Kumarasamy, "Link Selection sub-option for the Relay Agent Information Option for DHCPv4", RFC 3527, April 2003.
- [RFC4030] Stapp, M. and T. Lemon, "The Authentication Suboption for the Dynamic Host Configuration Protocol (DHCP) Relay Agent Option", RFC 4030, March 2005.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, December 2005.

9.2. Informative References

- [I-D.ietf-dhc-l2ra]
Joshi, B. and P. Kurapati, "Layer 2 Relay Agent Information", draft-ietf-dhc-l2ra-04 (work in progress), April 2009.
- [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.

Authors' Addresses

Ted Lemon
Nominum, Inc.
2000 Seaport Blvd
Redwood City, CA 94063
USA

Phone: +1 650 381 6000
Email: mellon@nominum.com

Hui Deng
China Mobile
53A, Xibianmennei Ave.
Beijing, Xuanwu District 100053
China

Email: denghui@chinamobile.com

Lu Huang
China Mobile
53A, Xibianmennei Ave.
Xunwu District, Beijing 100053
China

Email: huanglu@chinamobile.com

Dynamic Host Configuration (DHC)
Internet-Draft
Intended status: BCP
Expires: May 3, 2012

J. Brzozowski
Comcast Cable Communications
J. Tremblay
Videotron Ltd.
J. Chen
Time Warner Cable
T. Mrugalski
ISC
October 31, 2011

DHCPv6 Redundancy Deployment Considerations
draft-ietf-dhc-dhcpv6-redundancy-consider-02

Abstract

This document documents some deployment considerations for those who wishing to use DHCPv6 to support their deployment of IPv6. Specifically, providing semi-redundant DHCPv6 services is discussed in this document.

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	3
2. Scope and Assumptions	3
2.1. Service provider model	4
2.2. Enterprise model	5
3. Protocol requirements	5
3.1. DHCPv6 Servers	5
3.2. DHCPv6 Relays	5
3.3. DHCPv6 Clients	6
4. Deployment models	6
4.1. Split Prefixes	6
4.2. Multiple Unique Prefixes	8
4.3. Identical Prefixes	10
5. Challenges and Issues	12
6. IANA Considerations	14
7. Security Considerations	14
8. Acknowledgements	14
9. References	14
9.1. Normative References	14
9.2. Informative References	15
Authors' Addresses	15

1. Introduction

To support the deployment of IPv6 redundancy and high availability are required for many if not all components. This document provides information specific to the proposed near term approach for deploying semi-redundant DHCPv6 services in advance of DHCPv6 server implementations that support a standards based failover or redundancy protocol.

2. Scope and Assumptions

This document specifies an interim architecture to provide a semi-redundant DHCPv6 solution before the availability of vendor or standard based solutions. The proposed architecture may be used in wide range of networks, two notable deployment models are discussed: service provider and enterprise network environments. The described architecture leverages only existing and implemented DHCPv6 standards. This document does not address a standards based solution for DHCPv6 redundancy. In the absence of a standards based DHCPv6 redundancy protocol and implementation, some analogies are loosely drawn with the DHCPv4 failover protocol for reference. Specific discussions related to DHCPv4 failover and redundancy is out of scope for this document. Reader interested in initial work being done in DHCPv6 failover is recommended to read [I-D.mrugalski-dhc-dhcpv6-failover-requirements].

Although DHCPv6 redundancy may be useful in a wide range of scenarios, they may be generalized for illustration purposes in the two aforementioned. The following assumptions were made with regards to the existing DHCPv6 infrastructure, regardless of the model used:

1. At least two DHCPv6 servers are used to service to the same clients, but the number of servers is not restricted.
2. Existing DHCPv6 servers will not directly communicate or interact with one another in the assignment of IPv6 addresses and configuration information to requesting clients.
3. DHCPv6 clients are instructed to run stateful DHCPv6 to request at least one IPv6 address. Configuration information and other options like a delegated IPv6 prefix may be also requested.
4. Clients requesting IPv6 addresses, prefixes, and or options care of DHCPv6 must recognize and honor the DHCPv6 preference option. Furthermore, the requesting clients must process DHCPv6 ADVERTISE messages per [RFC3315] when the preference option is present.

5. DHCPv6 server failure does not imply failure of any other network service or protocol, e.g. TFTP servers. Redundancy of any additional services configured by means of DHCPv6 are outside of scope of this document. For example, a single DHCPv6 server may configure multiple TFTP servers, with preference for each TFTP server, as specified in [RFC5970].

While techniques described in this document provide some aspects of redundancy, it should be noted that complete redundancy will not be available until DHCPv6 protocol is standardized. Initial work toward that goal is described in [I-D.mrugalski-dhc-dhcpv6-failover-requirements].

2.1. Service provider model

The service provider model represents cases, where end-user devices may be configured directly, without any intermediate devices (like home routers used in service provider model). DHCPv6 clients include cable modems, customer gateways or home routers, and end-user devices. In some cases hosts may be configured directly using the service provider DHCPv6 infrastructure or via intermediate router, that is in turn being configured by the provider DHCPv6 infrastructure. The service provider DHCPv6 infrastructure may be semi-redundant in either case. Cable modems, customer gateways or home routers, and end-user devices are commonly referred to as CPE (Customer Premises Equipment). The following additional assumptions were made, besides the ones made in Section 2:

1. The service provider edge routers and access routers (CMTS for cable or DSLAM/BRAS for DSL for example) are IPv6 enabled when required.
2. CPE devices are instructed to perform stateful DHCPv6 to request at least one IPv6 address, delegated prefix, and or configuration information. CPE devices may also be instructed to leverage stateless DHCPv6 [RFC3736] to acquire configuration information only. This assumes that IPv6 address and prefix information has been acquired using other means.
3. The primary application of this BCP is for native IPv6 services. Use and applicability to transition mechanisms is out of scope for this document.
4. CPE devices must implement a stateful DHCPv6 client [RFC3315], support for DHCPv6 prefix delegation [RFC3633] or stateless DHCPv6 [RFC3736] may also be implemented.

2.2. Enterprise model

The enterprise model represents cases where end-user devices are most often configured directly without any intermediate devices (like home routers used in service provider model). However enterprise IPv6 environments quite often use or require that DHCPv6 relay agents are in place to support the use of DHCPv6 for the acquisition of IPv6 addresses and or configuration information. The assumptions here extend those that are defined in the beginning of Section 2:

1. DHCPv6 clients are hosts and are considered end nodes. Examples of such clients include computers, laptops, and possibly mobile devices.
2. DHCPv6 clients generally do not require the assignment of an IPv6 prefix delegation and as such do not support DHCPv6 prefix delegation [RFC3633].

3. Protocol requirements

The following sections outline the requirements that must be satisfied by DHCPv6 clients, relays, and servers to ensure the desired behavior is provided using pre-existing DHCPv6 server implementations as is. The objective is to provide a semi-redundant DHCPv6 service to support the deployment of IPv6 where DHCPv6 is required for the assignment of IPv6 addresses, prefixes, and or configuration information.

3.1. DHCPv6 Servers

This interim architecture requires DHCPv6 servers that are [RFC3315] compliant and support the necessary options required to support this solution. Essential to the the use of the interim architecture is support for stateful DHCPv6 and the DHCPv6 preference option both which are specified in [RFC3315]. For deployment scenarios where IPv6 prefix delegation is employed DHCPv6 servers must support DHCPv6 prefix delegation as defined by [RFC3633]. Further, where stateless DHCPv6 is used support for [RFC3736] is required by DHCPv6 servers.

3.2. DHCPv6 Relays

There are no specific requirements regarding relays. However, it is implied that DHCPv6 relay agents must be [RFC3315] compliant and must support the ability to relay DHCPv6 messages to more than one destination minimally.

3.3. DHCPv6 Clients

DHCPv6 clients are required to be compliant to [RFC3315] and support the necessary options required to support this solution depending on the mode of operations and desired behavior. Where prefix delegation is required DHCPv6 clients will be required to support DHCPv6 prefix delegation as defined in [RFC3633]. Clients used with this semi-redundant DHCPv6 deployment model must support the acquisition of at least one IPv6 address and configuration information using stateful DHCPv6 as specified by [RFC3315]. The use of stateless DHCPv6 which is also specified in [RFC3315] may also be supported. DHCPv6 client must recognize and adhere to the processing of the advertised DHCPv6 preference options sent by the DHCPv6 servers.

4. Deployment models

At the time of this writing a standards-based DHCPv6 redundancy protocol and implementations are not available. As a result DHCPv6 server implementations will be used as-is to provide best effort, semi-redundant DHCPv6 services. Behavior of the DHCPv6 services will in part be governed by the configuration used by each of the servers. Additionally, various aspects of the DHCPv6 protocol [RFC3315] will be leveraged to yield the desired behavior. No inter-server or inter-process communications will be used to coordinate DHCPv6 events and or activities. DHCP services for both IPv4 and IPv6 may operate simultaneously on the same physical server(s) or may operate on different ones.

4.1. Split Prefixes

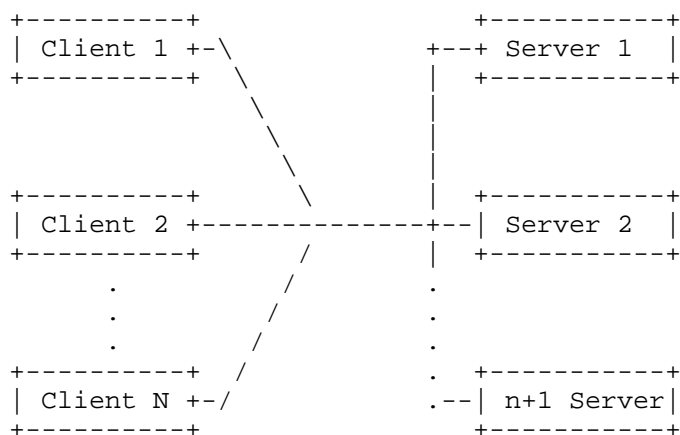
In the split prefixes model, each DHCPv6 server is configured with a unique, non-overlapping range derived from the /64 prefix deployed for use within an IPv6 network. Distribution between two servers, for example, would require that an allocated /64 be split in two /65 ranges. 2001:db8:1:0001:0000::/65 and 2001:db8:1:0001:8000::/65 would be assigned to each DHCPv6 server for allocation to clients derived from 2001:db8:1:0001::/64 prefix.

Each DHCP server allocates IPv6 addresses from the corresponding ranges per device class. Each DHCPv6 server will be simultaneously active and operational. Address allocation is governed largely through the use of the DHCPv6 preference option, so server with higher preference value is always preferred. Additional proprietary mechanisms can be leveraged to further enforce the favoring of one DHCP server over another. Example of such scenario is presented in Figure 1.

It is important to note that over time, it is possible that bindings may be disproportionally distributed amongst DHCPv6 servers and not any one server will be authoritative for all bindings.

Per [RFC3315], a DHCPv6 ADVERTISE messages with a preference option of 255 is an indicator to a DHCPv6 client to immediately begin a client-initiated message exchange by transmitting a REQUEST message. Alternatively, a DHCPv6 ADVERTISE messages with a preference option of any value lesser than 255 or absent preference option is an indicator to the client that it must wait for subsequent ADVERTISE messages before proceeding, as defined in Section 17.1.2 of [RFC3315]. Additionally, in the event of a DHCPv6 server failure it is desirable for a server other than the server that originally responded to be able to rebind the client. It is not critical, that the DHCPv6 server be able to rebind the client in this scenario, however, this is generally desirable behavior. Given the proposed architecture, the remaining active DHCPv6 server will have a different range configured making it technically incorrect for the same to rebind the client in its current state. Ultimately, when rebinding fails the client will acquire a new binding from the configured range unique to an active server. Furthermore, shorter T1, T2, valid, and preferred lifetimes can be used to reduce the possibility that a client or some other element on the network will experience a disruption in service or access to relevant binding data. The values used for T2, preferred and valid lifetime can be adjusted or configured to minimize service disruption. Ideally T2, preferred and valid lifetimes that are equal or near equal can be used to trigger a DHCPv6 client to reacquire IPv6 address, prefix, and or configuration information almost immediately after rebinding fails. It is important to note that shorter values will most certainly create additional load and processing for the DHCPv6 server, which must be considered.

Using a split prefix configuration model dynamic updates to DNS can be coordinated to ensure that the DNS is properly updated with current binding information. Challenges arise with regards to the update of PTR for IPv6 addresses since the DNS may need to be overwritten in a failure condition. The use of a split prefixes enables the differentiation of bindings and binding timing to determine which represents the current state. This becomes particularly important when DHCPv6 Leasequery [RFC5007] and/or DHCPv6 Bulk Leasequery [RFC5460] are leveraged to determine lease or binding state. An additional benefit is that the use of separate ranges per DHCPv6 server makes failure conditions more obvious and detectable.



```

Server 1
=====
Prefix=2001:db8:1:0:0::/64
Range=2001:db8:1:0:0::/65
Preference=255
    
```

```

Server 2
=====
Prefix=2001:db8:1:0:0::/64
Range=2001:db8:1:0:8000::/65
Preference=0
    
```

```

Server n+1
=====
Prefix, range, and preference would
vary based on range definition
    
```

Split prefixes approach.

Figure 1

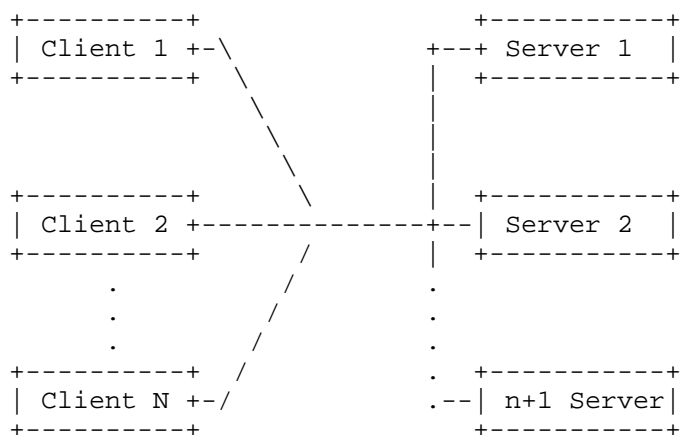
4.2. Multiple Unique Prefixes

In the multiple prefix model, each DHCPv6 server is configured with a unique, non-overlapping prefix. A /64 range equal to the prefix is configured on each server. For example, the range 2001:db8:1:0000::/64 would be assigned to a single DHCPv6 server for allocation to clients equal to its parent prefix 2001:db8:1:0000::/64. Subsequently the second DHCPv6 server could use 2001:db8:1:0001::/64 as range and prefix. This would be repeated for each active DHCP server. Example of this scenario is presented in Figure 2.

This approach uses a unique prefix and ultimately range per DHCPv6 server with corresponding prefixes configured for use in the network. The corresponding network infrastructure must in turn be configured to use multiple prefixes on the interface(s) facing the DHCPv6 client. The configuration is similar on all the servers, but a different prefix and a different preference is used per DHCPv6 server.

This approach would drastically increase the rate of consumption of IPv6 prefixes and would also yield operational and management challenges related to the underlying network since a significantly higher number of prefixes would need to be configured and routed. This approach also does not provide a clean migration path to the desired solution leveraging a standards-based DHCPv6 redundancy or failover protocol, which of course has yet to be specified.

The use of multiple unique prefixes provides benefits similar to those referred to in Section 4.1 related to dynamic updates to DNS. The use of multiple unique prefixes enables the differentiation of bindings and binding timing to determine which represents the current state. This becomes particularly important when DHCPv6 Leasequery [RFC5007] and/or DHCPv6 Bulk Leasequery [RFC5460] are leveraged to determine lease or binding state. The use of separate prefixes and ranges per DHCPv6 server makes failure conditions more obvious and detectable.



```

Server 1
=====
Prefix=2001:db8:1:0000::/64
Range=2001:db8:1:0000::/64
Preference=255
    
```

```

Server 2
=====
Prefix=2001:db8:1:1000::/64
Range=2001:db8:1:1000::/64
Preference=0
    
```

```

Server 3
=====
Prefix=2001:db8:1:2000::/64
Range=2001:db8:1:2000::/64
Preference=(>0 and <255)
    
```

Multiple unique prefix approach.

Figure 2

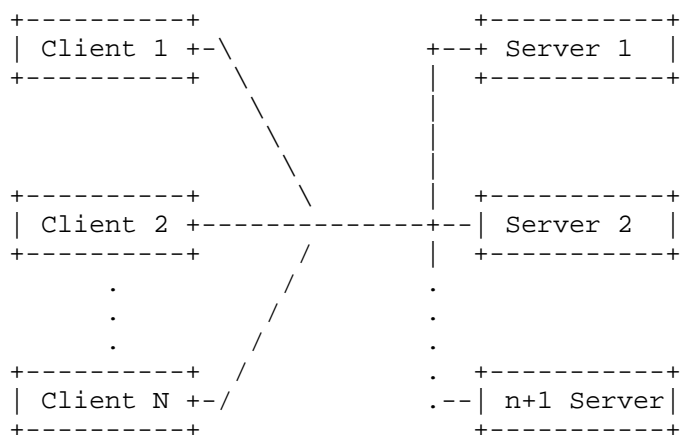
4.3. Identical Prefixes

In the identical prefix model, each DHCPv6 server is configured with the same overlapping prefix and range deployed for use within an IPv6 network. Distribution between two or more servers, for example, would require that the same /64 prefix and range be configured on all DHCP servers. For example, the range 2001:db8:1:0001:0000::/64 would be assigned to all DHCPv6 server for allocation to clients derived from 2001:db8:1:0001::/64 prefix. This would be repeated for each active DHCP server. Example of such scenario is presented in

Figure 3.

This approach uses the same prefix, length, and range definition across multiple DHCPv6 servers. All other configuration remaining the same the only other attribute of configuration option configured differently per DHCPv6 server would be DHCPv6 preference. This approach conceivably eases the migration of DHCPv6 services to fully support a standards based redundancy or failover protocol. Similar to the split prefix architecture described above this approach does not place any additional addressing requirements on network infrastructure.

The use of identical prefixes provides no benefit or advantage related to dynamic DNS updates, support of DHCPv6 Leasequery [RFC5007] or DHCPv6 Bulk Leasequery [RFC5460]. In this case all DHCP servers will use the same prefix and range configurations making it less obvious that a failure condition or event has occurred.



```

Server 1
=====
Prefix=2001:db8:1:0000::/64
Range=2001:db8:1:0000::/64
Preference=255
    
```

```

Server 2
=====
Prefix=2001:db8:1:0000::/64
Range=2001:db8:1:0000::/64
Preference=0
    
```

```

Server 3
=====
Prefix=2001:db8:1:0000::/64
Range=2001:db8:1:0000::/64
Preference=(>0 and <255)
    
```

Identical prefix approach.

Figure 3

5. Challenges and Issues

The lack of interaction between DHCPv6 servers introduces a number of challenges related to the operations of the same in a production environment. The following areas are of particular concern:

- o In identical prefixes scenario, both servers must follow the same address allocation procedure, i.e. they both must use the same algorithm and the same policy to determine which address is going

to be assigned to a specific client. Otherwise there is a distinct chance that each server will assign the same address to two different clients.

- o Interactions with DNS server(s) to support the dynamic update of the same address when one or more DHCPv6 servers have become unavailable. This specifically becomes a challenge when or if nodes that were initially granted a lease:
 1. Attempt to renew or rebind the lease originally granted, or
 2. Attempt to obtain a new lease

DHCID Resource Record, defined in [RFC4701], allows identification of the current owner for specific DNS data that can be used during DNS Update procedure [RFC2136]. [RFC4704] specifies how DHCPv6 servers and/or client may perform updates. [RFC4703] provides a way how to solve conflicts between clients. Although it deals with most cases, it is still possible to leave abandoned RR records. Consider following scenario. There are two independent servers. Server A assigns a lease to a client and updates DNS with AAAA record for assigned address and name. When the client renews, server A is not available and server B assigns a different lease. DNS is again updated (now two AAAA RRs are in the DNS for the client). Anyone trying to use the DNS information doesn't know which of the two leases is active. And, if server A never recovers, its information may never be removed.

- o Interactions with DHCPv6 servers to facilitate the acquisition of IPv6 lease data care of the DHCPv6 Leasequery [RFC5007] or DHCPv6 Bulk Leasequery [RFC5460] protocols when one or more DHCPv6 servers have become unavailable and have granted leases to DHCPv6 clients. If IPv6 lease data is required and the granting server is unavailable it will not be possible to obtain any information about leases granted until one of the following has taken place.
 1. The granting DHCPv6 server becomes available with all lease information restored
 2. The client has renewed or rebound its lease against a different DHCPv6 server

It is important to note that with DHCPv6 until such time that a redundancy or failover protocol is available binding updates and synchronization will not occur between DHCPv6 servers.

6. IANA Considerations

IANA is not requested to assign any numbers at this time.

7. Security Considerations

Security considerations specific to the operation of the DHCPv6 protocol are created through the use of this interim architecture for DHCPv6 redundancy beyond what has been cited for Dynamic Host Configuration Protocol for IPv6 (DHCPv6) [RFC3315]. There are considerations related to DNS, specifically the dynamic updating of DNS, when such models are employed. Potential opportunities are created to overwrite valid DNS resource records when provisions have been made accommodate some of the models cited in this document. In some cases this is desirable to ensure that DNS remains up to date when using one or more of these models, however, abuse of the same could result in undesirable behavior.

8. Acknowledgements

Many thanks to Bernie Volz, Kim Kinnear, Ralph Droms, David Hankins and Chuck Anderson for their input and review.

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).

9. References

9.1. Normative References

- [RFC2136] Vixie, P., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, April 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.

- [RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", RFC 3736, April 2004.
- [RFC4701] Stapp, M., Lemon, T., and A. Gustafsson, "A DNS Resource Record (RR) for Encoding Dynamic Host Configuration Protocol (DHCP) Information (DHCID RR)", RFC 4701, October 2006.
- [RFC4703] Stapp, M. and B. Volz, "Resolution of Fully Qualified Domain Name (FQDN) Conflicts among Dynamic Host Configuration Protocol (DHCP) Clients", RFC 4703, October 2006.
- [RFC4704] Volz, B., "The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Client Fully Qualified Domain Name (FQDN) Option", RFC 4704, October 2006.
- [RFC5007] Brzozowski, J., Kinnear, K., Volz, B., and S. Zeng, "DHCPv6 Leasequery", RFC 5007, September 2007.
- [RFC5460] Stapp, M., "DHCPv6 Bulk Leasequery", RFC 5460, February 2009.
- [RFC5970] Huth, T., Freimann, J., Zimmer, V., and D. Thaler, "DHCPv6 Options for Network Boot", RFC 5970, September 2010.

9.2. Informative References

- [I-D.mrugalski-dhc-dhcpv6-failover-requirements]
Mrugalski, T. and K. Kinnear, "DHCPv6 Failover Requirements",
draft-mrugalski-dhc-dhcpv6-failover-requirements-00 (work
in progress), June 2011.

Authors' Addresses

John Jason Brzozowski
Comcast Cable Communications
1306 Goshen Parkway
West Chester, PA 19380
USA

Phone: +1-609-377-6594
Email: john_brzozowski@cable.comcast.com

Jean-Francois Tremblay
Videotron Ltd.
612 Saint-Jacques
Montreal, Quebec H3C 4M8
Canada

Email: jf@jftremblay.com

Jack Chen
Time Warner Cable
13820 Sunrise Valley Drive
Herndon, VA 20171
USA

Email: jack.chen@twcable.com

Tomasz Mrugalski
Internet Systems Consortium, Inc.
950 Charter St.
Redwood City, CA 94063
USA

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

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 13, 2012

W. Dec, Ed.
Cisco Systems
T. Mrugalski
ISC
T. Sun
China Mobile
B. Sarikaya
Huawei USA
September 10, 2011

DHCPv6 Route Options
draft-ietf-mif-dhcpv6-route-option-03

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 March 13, 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	3
2. Problem overview	3
3. DHCPv6 Based Solution	4
3.1. Default route configuration	4
3.2. Configuring on-link routes	5
3.3. Deleting obsolete route	5
3.4. Applicability to routers	5
3.5. Updating Routing Information	5
3.6. Limitations	6
4. DHCPv6 Route Options	7
4.1. Next Hop Option Format	7
4.2. Route Prefix Option Format	8
5. DHCPv6 Server Behavior	10
6. DHCPv6 Client Behavior	10
7. IANA Considerations	11
8. Security Considerations	11
9. Contributors and Acknowledgements	12
10. References	12
10.1. Normative References	12
10.2. Informative References	12
Authors' Addresses	13

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 allows network administrators to better 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. 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.

3.1. Default route configuration

Defined mechanism may be used to configure default route. Default route may be specified in two ways.

In bandwidth constrained networks, server MAY send NEXT_HOP option without any RT_PREFIX options. NEXT_HOP option that does not contain any RT_PREFIX options designate default router. Second way of defining default route is to convey RT_PREFIX option that specifies ::/0 route, included as suboption in NEXT_HOP. First approach has the benefit of consuming less bandwidth, while the second one allows definition of default route lifetime and metric.

Server MUST NOT define more than one default prefix (i.e. both defined configuration methods are mutually exclusive). Unless there are significant bandwidth restrictions, mechanism that uses `::/0` RT_PREFIX option SHOULD be used.

3.2. 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.

3.3. 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.

3.4. 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.

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.

3.5. 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 RECONFIRGURE message to each client, thus forcing it to immediately start renewal process.

See also Section 3.6 about limitations regarding dynamic routing.

3.6. 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 interverntion 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]).

4. 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].

4.1. 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.

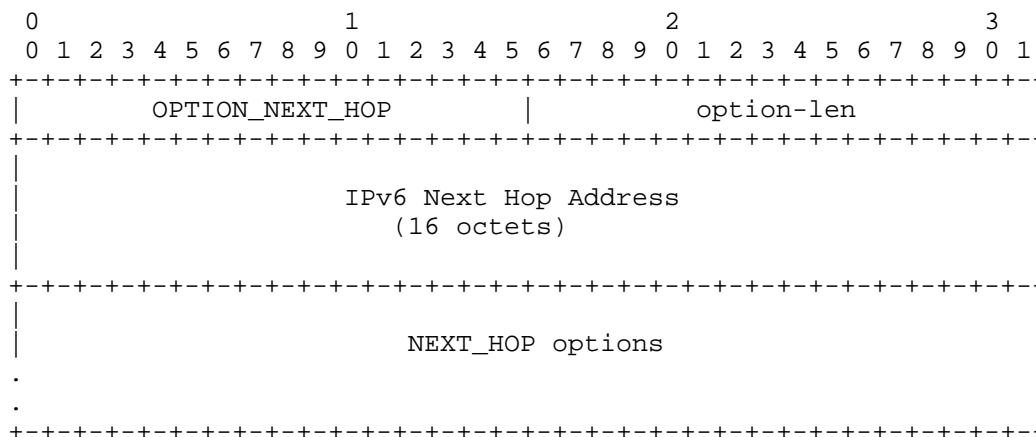


Figure 1: IPv6 Next Hop Option Format

option-code: OPTION_NEXT_HOP (TBD).

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.

4.2. 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.

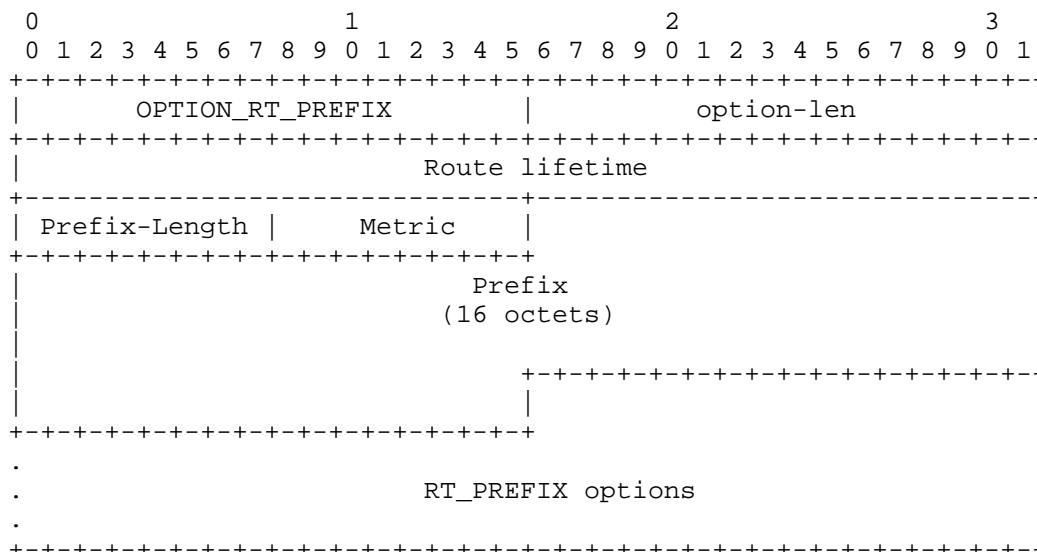


Figure 2: Route Prefix Option Format

option-code: OPTION_RT_PREFIX (TBD).

option-len: 18 + length of RT_PREFIX options.

Route lifetime 32-bit unsigned integer. Specifies lifetime of the route information, expressed in seconds. There are 2 special values defined. 0 means that route is no longer valid and must be removed by clients. 0xffffffff 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.

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: Fixed length 16 octet field containing an IPv6 prefix.

RT_PREFIX options: Options specific to this particular prefix.

5. 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]

6. 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. IANA Considerations

A DHCPv6 option number of TBD for the introduced Route Option. IANA is requested to allocate three DHCPv6 option codes referencing this document: OPTION_NEXT_HOP and OPTION_RT_PREFIX.

8. 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 6. 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 3.6 for details.

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

9. 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 and Alexandru Petrescu for their comments and useful suggestions.

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.
- [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.

10.2. Informative References

- [I-D.ietf-dhc-secure-dhcpv6]
Jiang, S. and S. Shen, "Secure DHCPv6 Using CGAs",
draft-ietf-dhc-secure-dhcpv6-03 (work in progress),

June 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-01 (work in progress), August 2011.
- [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.

Authors' Addresses

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

Email: wdec@cisco.com

Tomasz Mrugalski
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:

Network Working Group
Internet-Draft
Updates: 3315 (if approved)
Intended status: Standards Track
Expires: April 25, 2012

T. Mrugalski
ISC
October 23, 2011

Requesting Suboptions in DHCPv6
draft-mrugalski-dhc-dhcpv6-suboptions-02

Abstract

DHCPv6 clients may use Option Request Option (ORO) defined in RFC3315 [RFC3315] to specify, which options they would like to have configured by DHCPv6 servers. Clients may also be interested in specific options that do not appear in DHCPv6 message directly (top-level options), but rather as nested options or sub-options (i.e. options conveyed within other options). This document clarifies how to use already defined ORO to request specific options within scopes other than top-level. This document updates RFC3315.

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 April 25, 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	3
2. Terminology	3
3. Suboption Request Procedure	3
4. Justification	4
5. DHCPv6 Client Behavior	5
6. DHCPv6 Server Behavior	6
7. IANA Considerations	6
8. Security Considerations	6
9. Acknowledgements	6
10. References	6
10.1. Normative References	6
10.2. Informative References	6
Author's Address	7

1. Introduction

There are 2 ways DHCPv6 client can inform a server about its intent to have an option configured. The first (mandatory) way is to send Option Request Option (ORO), defined in [RFC3315]. The second way (optional, can be used as an addition to the first method) is to send the actual requested option to provide hints to a server.

Clients may also be interested in receiving specific sub-options (i.e. options that do not appear in DHCPv6 messages directly, but rather within other options). Unfortunately, there is no clear way for clients to request such sub-options. [RFC3315] does not provide any guidance regarding such problem. This document clarifies how clients should request sub-options.

2. Terminology

This section defines terms used in this document.

Option - Any DHCPv6 Option, defined according to format specified in [RFC3315]. Option may appear in DHCPv6 message directly or within other options.

Top-Level Option - an option that appears in DHCPv6 directly. Most existing options are top-level options.

Sub-Option - An option that appears not as top-level option, but rather within other option. An example of such option is IAADDR that may only appear within IA_NA or IA_TA options. Sub-options are sometimes referred to as nested options.

Scope - Any place (message or option) that is allowed to convey DHCPv6 options. Examples of scope are top-level (options conveyed directly within DHCPv6 message), IA_NA (options conveyed within specific instance of IA_NA option), or IA_PD (options conveyed within specific instance of IA_PD option).

3. Suboption Request Procedure

Clients that want specific option provided by the server, SHOULD include ORO within requested scope. This ORO MUST include requested option type. For example, if client expects to have suboption FOO configured in IA_NA, it should transmit IA_NA option that contains ORO. This ORO should convey a FOO option code and possibly other options requested within that scope.

Client MAY include several instances of ORO, one for each scope.
Client MUST NOT include more than one ORO in each scope.

Discussion: Aforementioned simple procedure is easy to implement, but it does not cover all cases. Therefore following extension may be taken into consideration.

There are cases, when client does not transmit options for each scope it expects to receive. Therefore client may not be able to follow procedure defined in previous section. In such case client SHOULD include ORO option in the inner-most scope that is closest to the location of desired option. For example, [I-D.ietf-dhc-pd-exclude] defines PD_EXCLUDE option that may be placed within IAPREFIX option, that in turn may be placed within IA_PD option that finally is placed in a DHCPv6 message. Client would like to receive PD_EXCLUDE option, but it in certain cases may choose to not send IAPREFIX within IA_PD, just empty IA_PD (e.g. in SOLICIT message). In such cases, client should include ORO within IA_PD, even though requested PD_EXCLUDE option will not be conveyed directly within IA_PD, but rather indirectly - within IAPREFIX that will be included in IA_PD.

Example: TODO (provide example of client requesting top-level and nested option, e.g. DNS_SERVER and PD_EXCLUDE).

4. Justification

As DHCPv6 protocol continues to be used to configure increasingly complex features, number of nested options will increase. To avoid each new document repeating the same sub-option request procedure, it seems reasonable to define such uniform procedure now. Even worse, such documents may propose different ways of requesting different options. This would considerably complicate server implementations.

Another alternative possible approach would be to simply use ORO as it is already defined. Client could include single instance of ORO to express desire to receive specific suboptions. Several existing server implementations deal with all options in a uniform way. Using top-level ORO to request suboptions would cause server to misplace requested options (i.e. to place them as top-level option rather than suboption). Avoiding such pitfalls, would complicate server implementation significantly, as servers would have to be configured with extra information regarding each option (where does specific option is supposed to appear - top level or as suboption). For example, in case when client requested PD_EXCLUDE and DNS_SERVERS options, server would have to handle each requested option differently and put one option inside an IAPREFIX option, while the other option directly in a message.

Discussion: (The following section should probably be removed if this draft is published). Currently there are several existing drafts that could benefit from this proposal:

1. [I-D.ietf-dhc-pd-exclude] defines PD_EXCLUDE option that is conveyed within IAPREFIX (that in turn is conveyed in IA_PD). Currently this draft calls for requesting PD_EXCLUDE in top-level ORO.
2. [I-D.ietf-mif-dhcpv6-route-option] defines a way to convey basic information about routers and prefixes available via those routers. It defines NEXT_HOP option that contains RT_PREFIX options. Each of those defined options may possibly convey additional, not yet defined routing related options, e.g. MTU, flow label, QoS parameters or many others.
3. There is at least one existing DHCPv6 implementation (Dibbler) that currently requests extra sub-options using top-level ORO.
4. A draft about configuring 4rd rules over DHCPv6 [I-D.mrugalski-dhc-dhcpv6-4rd] defines nested DHCPv6 options. Although this is early phase of the work and its layout will likely change (there is ongoing work within Softwires group on MAP that will likely include this work), the generic high level approach will remain similar. 4rd and MAP architectures require configuring one or more mapping rules. Each mapping rule consists of several mandatory (Domain IPv6 Prefix, Domain 4rd/MAP Prefix, Length of CE IPv6 Prefix) and one optional (Domain IPv6 suffix) parameters. As all those options are dedicated to configuration of different aspects of the same feature (4rd or MAP), there's distinct possibility that it will be defined as several options nested within a single grouping option. Although this architecture is a new proposal, there may be new extensions proposed, similar to extensions to DS-Lite architecture. This may result in potential new options related to 4rd/MAP.
5. DHCPv6 Client Behavior

In addition to standard behavior defined in [RFC3315] client SHOULD include ORO in each option that it would like to receive suboptions in. For example, if client wants to receive suboption FOO in IA_NA option, it SHOULD transmit IA_NA option that contains a single ORO with FOO option code.

6. DHCPv6 Server Behavior

Server processes the message received from client in a regular way, as explained in [RFC3315]. For each option that is allowed to have suboptions (i.e. for each scope), server checks if there is ORO present. For each ORO present, server appends requested options if it is configured to do so.

7. IANA Considerations

IANA is not requested to take any actions regarding this document.

8. Security Considerations

TBD

9. Acknowledgements

Author would like to thank Bernie Volz, Jouni Korhonen, Ole Troan and Ted Lemon for their insightful comments.

This work has been partially supported by Department of Computer Communications (Gdansk University of Technology) and by 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).

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.
- [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.

10.2. Informative References

- [I-D.ietf-dhc-pd-exclude] Korhonen, J., Savolainen, T., Krishnan, S., and O. Troan, "Prefix Exclude Option for DHCPv6-based Prefix Delegation", draft-ietf-dhc-pd-exclude-03 (work in progress)

progress), August 2011.

[I-D.ietf-mif-dhcpv6-route-option]

Dec, W., Mrugalski, T., Sun, T., and B. Sarikaya, "DHCPv6 Route Options", draft-ietf-mif-dhcpv6-route-option-03 (work in progress), September 2011.

[I-D.mrugalski-dhc-dhcpv6-4rd]

Mrugalski, T., "DHCPv6 Options for IPv4 Residual Deployment (4rd)", draft-mrugalski-dhc-dhcpv6-4rd-00 (work in progress), July 2011.

Author's Address

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

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

