DHCPv6 class based prefix
draft-bhandari-dhc-class-based-prefix-01

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 September 13, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents

Table of Contents

1. Introduction ................................................. 3
   1.1. Motivation .............................................. 3
   1.2. Terminology ............................................. 4
   1.3. Requirements Language ................................. 4
2. Overview ..................................................... 4
   2.1. Prefix Class Option in IA_PD ......................... 4
2.2. Consideration for different DHCPv6 entities .............. 5
   2.2.1. Requesting Router Behavior .......................... 5
   2.2.2. Delegating Router Behavior .......................... 6
   2.2.3. DHCPv6 Client Behavior for IA_NA allocation ....... 7
2.3. Usage .................................................... 7
   2.3.1. Class based prefix and IA_NA allocation .......... 7
   2.3.2. Class based prefix and IA_PD allocation .......... 8
   2.3.3. Class based prefix and SLAAC ....................... 8
3. Example Application ........................................... 11
   3.1. Class based prefix delegation ......................... 11
   3.2. IPv6 address assignment from class based prefix ....... 11
4. Acknowledgements ............................................ 12
5. IANA Considerations ......................................... 12
6. Security Considerations ..................................... 12
7. Change History (to be removed prior to publication as an
   RFC) ....................................................... 12
8. References .................................................. 12
   8.1. Normative References ................................. 12
   8.2. 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. 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.

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 referred to as "Prefix Color" in the mobile networks. 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 tunnel 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.

1.2. 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_PREFIX_CLASS        |          option-length    |
+---------------+---------------+---------------+---------------+---------------+---------------+---------------+
|                        prefix-class                           |
(variable length)                                          |
+---------------+---------------+---------------+---------------+---------------+---------------+---------------+
```

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:

- Configuration
- Prefix delegated via a DHCPv6 request to another DHCPv6 server
- 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:

- 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:

- It MAY have a pre-configured mapping between the prefix class and OPTION_USER_CLASS option received in IA_NA.

- It MAY match the OPTION_PREFIX_CLASS if the IA_NA request received contains OPTION_PREFIX_CLASS.

- It MAY map OPTION_PREFIX_CLASS option to the OPTION_USER_CLASS option by string matching of both these option values.
It MAY have a pre-configured mapping between the prefix class and the client DUID received in DHCPv6 message.

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 mobile network. Each of the examples will refer to the below network:

The example network consists of:

Mobile Gateway It is network entity anchoring IP traffic in the mobile core network. This entity allocates an IP address which is topologically valid in the mobile network and may act as a mobility anchor if handover between mobile and Wi-Fi is supported.

Mobile Nodes (MN) A host or router that changes its point of attachment from one network or subnetwork to another. A mobile node may change its location without changing its IP address; it may continue to communicate with other Internet nodes at any location using its (constant) IP address, assuming link-layer connectivity to a point of attachment is available.

Access Point (AP) A wireless access point, identified by a MAC address, providing service to the wired network for wireless nodes.
Access Router (AR) An IP router residing in an access network and connected to one or more Access Point (AP)s. An AR offers IP connectivity to MNs.

WLAN controller (WLC) The entity that provides the centralized forwarding, routing function for the user traffic.
Example mobile network

![Diagram of a mobile network with various operators, gateways, and access points, illustrating the flow of packets from one network to another.](image)

Figure 1
3.1. Class based prefix delegation

The Access Aggregation Gateway requests for Prefix delegation from Mobile gateway and associates the prefix received with prefix class "global-anchor". The Access Aggregation Gateway is preconfigured to provide prefixes from the following classes: "global-anchor", "local-breakout", "guest". It has a preconfigured policy to advertise prefixes to requesting routers and mobile nodes based on the service class supported by the service provider for the requesting device.

In the example mobile network, the Access Router(AR) requests class based prefix allocation by sending a DHCPv6 SOLICIT message and include OPTION_PREFIX_CLASS in the OPTION_ORO.

The Access Router (AR) receives an advertise with following prefixes in the IA_PD option:

1. P1: IA_PD Prefix option with a prefix 3001::1::/64 containing OPTION_PREFIX_CLASS set to "global-anchor"
2. P2: IA_PD Prefix option with a prefix 3001::2::/64 containing OPTION_PREFIX_CLASS set to "local-breakout"
3. P3: IA_PD Prefix option with a prefix 3001::3::/64 containing OPTION_PREFIX_CLASS set to "guest"

It sends a REQUEST message with all of above prefixes and receives a REPLY message with prefixes allocated for each of the requested class.

3.2. IPv6 address assignment from class based prefix

When the Access Router(AR) receives a DHCPv6 SOLICIT requesting IA_NA from the mobile node that has mobility service enabled, it offers an IPv6 address from the prefix class "global-anchor". For MN3 it advertises 3001::1::1 as the IPv6 address in OPTION_IAADDR in response to the IA_NA request.

The Mobile Node(MN4) Figure 1 sends a DHCPv6 SOLICIT message requesting IA_NA address assignment with OPTION_USER_CLASS option containing the value "guest" towards the CPE. The Access Router(AR) assumes the role of the DHCPv6 server and sends an ADVERTISE to the MN with OPTION_IA_NA containing an IPv6 address in OPTION_IAADDR from the "guest" prefix class. The IPv6 address in the OPTION_IAADDR is set to 3001::3::1. The "guest" class can also be distinguished based on a preconfigured interface or SSID advertised for MNs connecting to it.

When the Access Aggregation Gateway receives a DHCPv6 SOLICIT 

requesting IA_NA from MNs through WLC and it has a preconfigured profile to provide both local-breakout internet access and global-anchor, it offers an IPv6 address from the prefix class "local-breakout" and "global-anchor". For MN1 it advertises 3001::2::1 and 3001::1::2 as the IPv6 address in OPTION_IAADDR in response to the IA_NA request. Applications within MN1 can choose to use the appropriate prefix based on the mobility enabled or local-breakout property.

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. Change History (to be removed prior to publication as an RFC)

Changes from -00 to -01

a. Modified motivation section to focus on mobile networks

b. Modified example with a mobile network and class based prefix delegation in it

8. References

8.1. Normative References

8.2. Informative References


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: denghui02@gmail.com
Security option extensions for DHCP
draft-bi-dhc-sec-option-02

Abstract

This document defines a new option that can be used by DHCP servers to provision with DHCP clients specific security configuration information. It has been known that DHCP protocol typically works at the very beginning stage of the access to networks, thus lack of security protection. However, although it is difficult to set up some security mechanism for DHCP protocol solely, it is able to play a key role for DHCP server to provide configuration information to help building security mechanism within those pre-configured DHCP clients and devices. This new option defines a simple extension to current standard format and benefits to those who need to activate security mechanism in an early stage and interoperate within devices from multiple vendors.

Status of this Memo

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

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

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

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

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

E. Bi et al. Expires January 17, 2012 [Page 1]
1 Introduction

DHCP provides a framework for passing network configuration information to hosts on a TCP/IP network. Some configuration parameters and control information can be carried in DHCP options which are defined in [RFC2132], [RFC3046], [RFC3118], [RFC4030], etc. When a host that acts as a DHCP client booting up, it can be configured with some security policy. Such as, due to the security concern, all the IP packets to and from a client may be required to be protected by a secure mechanism, which is typically an IPsec channel or transport layer security established with the server or administrator.

These security mechanisms require the configuration information can be provisioned to the DHCP client at the early stage when it is connected to the network. In particular, the DHCP client should be notified the crucial security configuration information as early as possible. DHCP is essential for users who want to connect to IP networks before they can communicate with other hosts. Among a number of indispensable Internet protocols, it provides the most convenient way to make configuration extension, which eliminates the manual task by a network administrator and duplicate resource assignments. Thus, in addition to the essential IP address and network boot servers, security configuration information is expected to be included in DHCP extension, as well.

1.1 Applicability

Some scenarios that require this kind of provisioning secure configuration information are when DHCP clients in wireless base stations are attaching to a wireless network infrastructure. As defined in [3GPP.33.310], for establishing security link with operator’s network, wireless base station shall connect to the PKI server and SeGW. If secure configuration information (address of PKI server, address of SeGW, etc.) is unavailable on the wireless base stations, the wireless base stations cannot connect to the network. So it is important for the host to obtain a set of security configuration information, which is configured in the DHCP server prior to the establishment of the security tunnel. Currently, some implementations exchange this security information through DHCP vendor-specific options, i.e. OPTION 43. However, this has the usual limitations of requiring the client and server to understand these vendor-specific extensions. Since most of the security configuration information are common across most clients and servers, having a standardized set of options and procedures would be a huge benefit to interoperability. This document defines a new security DHCP option used to exchange the security configuration information.
The newly defined option is as follows:

Option: DHCP security specific configuration option

2 Terminology Used in This Document

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

3 DHCP Security Specific Configuration Option

A DHCP server can use this option to indicate to the DHCP client specific configuration information, such as the address of the security gateway that is used to establish IPsec tunnel within the enterprise network, the address of the PKI server which is used to issue certificate to the DHCP client.

This option may be used wherever DHCP options are available, as specified in [RFC2131] and [RFC2132]. It is most meaningful in the messages between DHCP client and DHCP server, such as, DHCPOFFER and DHCPACK. The format of the option is as follows:

```
Opt-ID | opt-length | attribute 1 | attribute 2 | ......
```

where Opt-ID denotes the new option ID, opt-length denotes the bit length of following attributes, and attributes 1,2... can be extended correspondingly to various use cases.

For example, in the above use case of 3GPP standards [3GPP.33.310], the addresses (IP or FQDN) of Se-GW and PKI server are needed to know for DHCP client (wireless base station), the DHCP option could be as follows, where attributes 1 and 2 denote Se-GW and PKI server, respectively.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Attribute 1 | data-len1 | Security-GW ID |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Security-GW IP Address Data |
|/|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Attribute 2 | data-len2 | PKI Server ID |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| PKI Server IP Address Data |
|/|
```

Figure 1. The format of DHCP security specific configuration option

This option contains the information corresponding to one or more security-specific code number. Multiple instances of this option may be present and must be concatenated in accordance with [RFC3396]. The definition of the information carried in this option is defined uniformly. The security-specific attribute information indicates the security information type. For example, a DHCP client indicates security configuration information with the same code number that can be interpreted by different DHCP servers. As the security-specific code is uniform and standard, no ambiguity interpretation can occur. A security-specific code number is unique and only occur once in the option and should be treated independently. This option can also contains one or more encapsulated options that defined in [RFC4361].

DHCP client can request the configuration information from DHCP server by sending DHCP request message. DHCP server allocates the configuration information to DHCP client according to the client ID. i.e., DHCP server can know whether this connecting client needs to be configured by client ID, which can be carried in option 60 specified in [RFC4361]. If the security configuration information is needed, the defined security-specific option will be sent back to the client from DHCP server in DHCPOFFER. If this option is used, different DHCP clients implemented by different vendors have good interoperability. The DHCP server needs only to support one standardized format which reduces complexity and enhances performance.

If the DHCP client is configured with a security policy, all of the attributes listed in the figure MUST be carried in the newly defined option in DHCPDISCOVER or DHCPREQUEST messages. And DHCP server MUST allocate all the requested configuration attributes according to the received attribute type and format in the DHCP response messages.

Use of security-specific information allows enhanced operation, utilizing additional features in a DHCP implementation. Servers not equipped to interpret the security-specific information sent by a client MUST ignore it. Clients that do not receive desired security-specific information MUST ignore it and initiate another DHCP operation.

Finally, it is also desired to extend this option to IPv6, which is left to be improved.
4 Security Considerations

This document defines a new security option used by DHCP servers and DHCP clients to provision security configuration information. However, the security mechanism itself does not need to rely on the security of DHCP, i.e. the configuration information provided by DHCP server can hardly guarantee their own validity, since DHCP originally is lack of protection.

Therefore, the new option proposed in this document is not aimed to solve the security of DHCP, but try to make use of DHCP to provide configuration information for those who have already equipped with security mechanism, and that security cannot be harmed by potentially invalid information provisioned.

5 IANA Considerations

There may be IANA consideration for taking additional value for the option. The value of the protocol field needed to be assigned from the numbering space.

6 Acknowledgements

Thanks to Eric Chen, Xiangsong Cui and Rock Xie who contributed actively to this document.

7 References

7.1 Normative References


7.2 Informative References

[3GPP.33.310] 3GPP, "Network Domain Security (NDS); Authentication Framework (AF)", 3GPP TS 33.310 10.3.0, June 2011.


Authors’ Addresses

Emily Bi
Huawei Technologies
Beijing, China
Email: bixiaoyu@huawei.com

Serge Manning
Huawei Technologies
TX, U.S.
Email: serge.manning@huawei.com

Marcus Wong
NJ, U.S.
Huawei Technologies
Email: mwong@huawei.com

Yang Cui
Huawei Technologies
Beijing, China
Email: cuiyang@huawei.com
Abstract

This document specifies the format and mechanism that is to be used for encoding client link-layer address in DHCPv6 messages by defining a new DHCPv6 Client Link-layer 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 September 13, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents.
carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction .................................................. 3
2. Problem Background and Scenario ........................... 3
3. DHCPv6 Client Link-layer 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. Change History (to be removed prior to publication as an RFC) ................................................. 5
11. Normative References ....................................... 6
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 link-layer
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,
effectively an enablement of dual-stack, there will be devices that
act as both DHCPv4 and DHCPv6 clients. In service provider
deployments, a typical DHCPv4 implemention 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 link-layer address in
the DHCPv6 message sent towards DHCPv6 Server. Similarly DHCPv6
Relay or Server cannot glean client link-layer 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 link-layer
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 link-layer
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
link-layer address in DHCPv6 will help in proving additional information in event debugging and logging related to the client at relay and server. The proposed option may be used in wide range of networks, two notable deployment models are service provider and enterprise network environments.

3. DHCPv6 Client Link-layer Address Option

The format of the DHCPv6 Client Link-layer Address option is shown below.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| OPTION_CLIENT_LINKLAYER_ADDR  |           option-length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    hardware type (16 bits)    |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|               link-layer address (variable length)            |
                                                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

option-code: OPTION_CLIENT_LINKLAYER_ADDR (TBD)
option-length: 2 + length of link-layer address
hardware type: Client Link-layer address type. The hardware type MUST be a valid hardware type assigned by the IANA, as described in [RFC0826]
link-layer address: Client Link-layer address.

4. DHCPv6 Client Behavior

All hosts or clients MAY include DHCPv6 Client link-layer address option in all the upstream DHCPv6 messages.

5. DHCPv6 Relay Agent Behavior

DHCPv6 Relay agents which are directly connected to clients/hosts MAY look for Client Link-layer Address option in the incoming DHCPv6 client message. Irrespective of the presence of client link-layer option in incoming DHCPv6 client messages, DHCPv6 Relay agents MAY include client link-layer address option in relayed DHCPv6 (RELAY-FORW) message. The DHCPv6 Relay agent behaviour can depend on configuration that decides whether Client Link-layer 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 link-layer address, it SHOULD first look for the client link-layer address option in the RELAY-FORW DHCP message of the DHCPv6 Relay agent closest to the client. In case it is not found, Server SHOULD look for client link-layer address option in the client DHCP message. Further, this behavior w.r.t the precedence of DHCPv6 server to look for Client link-layer address option can be overridden based upon the local policies.

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_LINKLAYER_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

Many thanks to Bernie Volz, Hemant Singh, Simon Hobson, Tina TSOU, Andre Kostur, Chuck Anderson, Steinar Haug, Niall O’Reilly, Jarrod Johnson and Vincent Zimmer for their input and review.

10. Change History (to be removed prior to publication as an RFC)

Changes from -00 to -01

a. "hardware address" has been renamed to "Link-layer address" to be consistent with DHCPv6 terminology
b. 1 byte chtype in DHCPv6 Client Link-layer Address option is replaced with 2 byte hardware type

c. chaddr in DHCPv6 Client Link-layer Address option is renamed as link-layer address

11. Normative References


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
Client Identifier Option in DHCP Server Replies
draft-ietf-dhc-client-id-02

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 September 13, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal
Provisions Relating to IETF Documents
(http://trustee.ietf.org/license-info) in effect on the date of
publication of this document. Please review these documents
carefully, as they describe your rights and restrictions with respect
to this document. Code Components extracted from this document must
include Simplified BSD License text as described in Section 4.e of
the Trust Legal Provisions and are provided without warranty as
described in the Simplified BSD License.

Table of Contents

1. Introduction ................................................. 3
2. Problem 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 within 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.

<table>
<thead>
<tr>
<th>Option</th>
<th>DHCPOFFER</th>
<th>DHCPACK</th>
<th>DHCPNAK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client identifier (if sent by client)</td>
<td>MUST</td>
<td>MUST</td>
<td>MUST</td>
</tr>
<tr>
<td>Client identifier (if not sent by client)</td>
<td>MUST NOT</td>
<td>MUST NOT</td>
<td>MUST NOT</td>
</tr>
</tbody>
</table>

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


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
DHCPv4 over IPv6 Transport
draft-ietf-dhc-dhcpv4-over-ipv6-01

Abstract

In IPv6 networks, there remains a need to provide IPv4 service for some residual devices. This document describes a mechanism for allocating IPv4 addresses to such devices using DHCPv4 with an IPv6 transport. It is done by extending DHCP client and server behavior, and by adding a new Relay Agent Information option to carry the IPv6 address of the client.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 13, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect
to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction ........................................... 3
2. Requirements Language .................................... 3
3. Terminology ............................................ 4
4. Protocol Summary ......................................... 4
5. Client Relay Agent IPv6 Address Sub-option ............... 6
6. Client Relay Agent Behavior .............................. 6
7. IPv6-Transport Server Behavior ........................... 7
8. IPv6-Transport Relay Agent Behavior ....................... 7
9. Security Consideration ................................... 8
10. IANA consideration ...................................... 8
11. References ............................................. 8
    11.1. Normative References ............................... 8
    11.2. Informative References ............................. 9
1. Introduction

DHCPv4 [RFC2131] was not designed with IPv6 in mind: DHCPv4 cannot operate on an IPv6 network. However, as dual-stack networks become a reality, the need arises to allocate IPv4 addresses in an IPv6 environment. To meet this demand, this document extends DHCPv4 to allow the use of an IPv6 network for transport.

A typical scenario that probably requires this feature is IPv4-over-IPv6 hub and spoke tunnel [RFC4925]. In this scenario, IPv4-over-IPv6 tunnel is used to provide IPv4 connectivity to end users (hosts or end networks) across an IPv6 network. If the IPv4 addresses of the end users are provisioned by the concentrator side, then the provisioning process should be able to cross the IPv6 network, too. One such tunnel mechanism is demonstrated in [I-D.ietf-softwire-public-4over6]. DHCPv4 over IPv6 would be a generic solution for this scenario.

Three main flavours of solutions may be considered:

- Use DHCPv6 instead of DHCPv4, to provision IPv4-related connectivity. In DHCPv6, the provisioned IPv4 address can be embedded into IPv6 address, or carried within a new option. Along with that, dedicated options are needed to convey IPv4-related information, such as the IPv4 address of DNS server, NTP server, etc. Therefore it will put a certain amount of IPv6-unrelated information into DHCPv6 protocol.

- Use DHCPv4 and tunnel DHCPv4-in-IPv4 messages over IPv6. Unlike the previous approach where DHCPv6 is used for both IPv4 and IPv6 connectivity, this approach consists in preserving the separation between IPv4 and IPv6 connectivity information. It allows to maintain the IPv4 service without major modification of IPv6-related provisioning resources, and sustains DHCPv4 to be the IPv4-related information carrier. However, this approach enforces an IPv4-in-IPv6 tunnel on DHCP, and requires extra efforts to maintain tunnel endpoint information for encapsulation use.

- Use DHCPv4 and extend it to work over IPv6 transport. Instead of relying on IPv4-in-IPv6 tunnel, this flavour uses IPv6 directly for DHCP message transport, and it keeps the advantage of separation with IPv6 connectivity information. This document focuses on this flavour. The document will define the extensions of DHCPv4 protocol behavior, as well as a new suboption of the Relay Agent Information Option, to fully support DHCPv4 over IPv6.

2. Requirements Language
3. Terminology

This document makes use of the following terms:

- **DHCPv4**: IPv4 Dynamic Host Configuration Protocol [RFC2131].

- **Client Relay Agent (CRA)**: a special DHCPv4 Relay Agent that sits on the same, IPv6-accessible host with the DHCPv4 client. CRA works as a "bridge" between DHCPv4 client and the IPv6 network, to convert between IPv4 transport and IPv6 transport.

- **IPv6-Transport Server (TSV)**: a DHCPv4 Server that supports IPv6 transport. TSV can listen on IPv6 for incoming DHCPv4 messages, and send DHCPv4 messages in IPv6 packets.

- **IPv6-Transport Relay Agent (TRA)**: a DHCPv4 Relay Agent that supports IPv6 transport. TRA sits on a machine which has both IPv6 and IPv4 connectivity, and relays DHCP messages between CRA and normal DHCPv4 server.

- **Client Relay Agent IPv6 Address Sub-option (CRA6ADDR suboption)**: a new suboption of DHCP Relay Agent Information Option [RFC3046] defined in this document. CRA6ADDR suboption is used by TRA to carry the IPv6 address of a CRA.

4. Protocol Summary

The scenario for DHCPv4 over IPv6 transport is shown in Figure 1. DHCPv4 clients and DHCPv4 server/relay are separated by an IPv6 network in the middle. DHCP messages between a client and the server/relay cannot naturally be forwarded to each other because they are by default IPv4 UDP packets, either unicast or broadcast. To bridge this gap, both the client side and the server/relay side should enable DHCPv4 over IPv6 transport. More precisely, they should support delivering and receiving DHCP messages in IPv6 UDP packets and thereby traverse the IPv6 network.

On the client side, a special relay agent called Client Relay Agent is placed on the same machine with the client. CRA is used to relay DHCP messages from the client to the server, and from the server to the client. CRA sends DHCPv4 messages to the server through unicast IPv6 UDP, and receives unicast IPv6 UDP packets with the DHCPv4 messages from the server. By using CRA, no extension is required on the DHCP client.
The IPv6-Transport DHCPv4 server can receive DHCP messages delivered in IPv6 UDP from CRA, and send out DHCP messages to CRA using IPv6 UDP (figure 2(a)). TSV should send DHCP messages to the IPv6 address from which it receives relevant DHCP messages earlier.

When CRAs communicate with an IPv6-Transport Relay Agent rather than with a server directly, the situation will become a little more complicated. Besides the IPv6 communication with CRA, TRA also communicates with a regular DHCPv4 server through IPv4. Therefore, when TRA relays DHCP messages between a CRA and the DHCPv4 server, it receives DHCP message from the CRA in IPv6 and sends it to the server in IPv4, while receives DHCP message from the server in IPv4 and sends it to the CRA in IPv6. TRA has to use the DHCP Relay Agent Information Option (Option 82) to record the IPv6 address of a CRA, which will be used as forwarding destination when relaying a DHCP message from the server. Since Option 82 doesn't have an existing suboption that fits in the case, this document defines a new Client Relay Agent IPv6 Address Sub-option.

---

5. Client Relay Agent IPv6 Address Sub-option

This suboption MUST be added by a DHCPv4 TRA. It encodes the IPv6 address of the host from which a DHCPv4-in-IPv6 CRA-to-TRA message was received. It is intended for the TRA to relay DHCPv4 replies back to the proper CRA. To be more specific, the TRA uses the IPv6 address encoded in this suboption as the destination IPv6 address when relaying a DHCPv4 reply to IPv6 network.

The CRA IPv6 address MUST be unique in the IPv6 domain.

The CRA6ADDR suboption has a fixed length of 18 octets. The SubOpt code is tbd by IANA, the length field should be 16, and the following 16 octets contain the CRA IPv6 address.

DHCP servers MAY use this suboption to select parameters specific to particular hosts. Servers MAY parse this suboption and extract the semantic of IPv6 address.

```
SubOpt   Len     Agent Remote ID
+--------+---------+-------------------+
| tbd    |  16     | a1    a2    a3    ... | a16 |
```

Figure 3 Client Relay Agent IPv6 Address Sub-option format

6. Client Relay Agent Behavior

A Client Relay Agent sits on the same machine with the DHCPv4 client. CRA is a special type of relay agent, which relays DHCPv4 messages between regular client and TSV/TRA. The communication between CRA and the client happens within the machine using IPv4, and the communication between CRA and TSV/TRA happens on the IPv6 network using IPv6.

A CRA is configured with one or more IPv6 addresses of TSV/TRA. This configuration is provided before DHCPv4 process, for example through DHCPv6 option, or by some other mechanisms depending on the application scenarios.

A CRA listens for DHCP messages on IPv4 UDP port 67. When it receives from IPv4 any DHCP message with bootp op field = 1, it forwards the message using the standard DHCP relay agent format, but over UDPv6, with source port 67 and destination port 67. Here the
CRA MUST NOT include an option 82 or modify the giaddr field of the DHCP message. The CRA forwards the message to each of the DHCP server or relay agent with which it is configured. The CRA MUST use a global IPv6 address if it has one.

A CRA also listens for DHCP messages on IPv6 UDP port 68. When it receives from IPv6 any DHCP message with bootp op field = 2, the CRA checks to see if the message contains option 82, and if so, it discards the message. Otherwise, it delivers the message to the DHCP client using IPv4.

7. IPv6-Transport Server Behavior

To support IPv6 transport, the behavior of DHCPv4 server should be extended. The IPv6-Transport Server can listen on IPv6 port 67 for DHCPv4 messages, and send DHCPv4 messages through IPv6.

A TSV listens for DHCP messages on IPv6 UDP port 67. When it receives from IPv6 a DHCP message, it MUST record the IPv6 source address of that message and retain it as the return address of the message. That is to say, when sometime later the TSV responds to this message, it MUST send the reply message to the IPv6 return address retained earlier, with destination port 68. When the TSV receives an DHCP message with a CRA6ADDR option, the TSV handles it following the standard option 82 procedure defined in [RFC3046]. The rest of TSV DHCP process is the same with normal DHCPv4 server. A TSV can also listen on IPv4 UDP port 67 like a normal DHCPv4 server, and process normally when receives IPv4 DHCPv4 message.

This document places no new requirements on DHCPv4 servers that do not listen on UDPv6—in order to use an IPv4-only DHCPv4 server through an IPv6 connection, a TRA is required.

8. IPv6-Transport Relay Agent Behavior

An IPv6-Transport Relay Agent sits between IPv6 network and IPv4 network, and relays DHCPv4 message between CRAs and IPv4-only DHCPv4 server. The communication between CRAs and the TRA uses IPv6, while the communication between TRA and server uses IPv4. A TRA listens on IPv6 UDP port 67 for DHCP messages with bootp op field = 1, as well as IPv4 UDP port 68 for DHCP messages with bootp op field = 2.

When relaying a DHCP message from CRA to server, TRA MUST add an option 82 with a CRA6ADDR suboption. This suboption contains the IPv6 source address of the message (the CRA’s IPv6 address) which is retained when the message is received in IPv6. The TRA MUST also store the IPv4 address of itself in the giaddr field of the DHCP message. The TRA MAY include a Link Selection Suboption [RFC3527] to
indicate to the DHCP server which link to use when choosing an IP address.

When receiving a DHCP message from the DHCP server, if the option 82 in the message contains no CRA6ADDR suboption, the TRA MUST discard the message. Otherwise, it processes it as required by [RFC3046], and forwards it to the IPv6 address recorded in the CRA6ADDR suboption, with source port 67 and destination port number 68. TRA SHOULD drop DHCPv4-over-IPv6 traffic that is not originated from configured server address.

9. Security Consideration

This mechanism may rise a new form of DHCP protocol attack. A malicious attacker in IPv6 can interfere with the DHCPv4 process by inject fake DHCPv4-in-IPv6 messages which will be handled by TSV or TRA. However, the damage is the same with the known DHCPv4 attack happened in IPv4. The only difference is the attacker and the victim could locate in different address families.

Another impact is DHCP filtering. There are firewalls today capable of filtering DHCP traffic (DHCPv4 over IPv4 and DHCPv6 over IPv6 packages). The DHCP messages with the new, DHCPv4-in-IPv6 style may bypass these firewalls. Nevertheless it is not difficult for them to make some slight modification and adapt to the new DHCPv4 message pattern.

10. IANA consideration

IANA is requested to 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 Client Relay Agent IPv6 Address Sub-option.

11. References

11.1. Normative References


Internet-Draft                  DHCPv4 over IPv6               March 2012


11.2.  Informative References


Authors’ Addresses

Yong Cui
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6260-3059
EMail: cuiyong@tsinghua.edu.cn

Peng Wu
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5822
EMail: pengwu.thu@gmail.com

Jianping Wu
Tsinghua University
Department of Computer Science, Tsinghua University
Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on March 17, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
Abstract

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) enables DHCPv6 servers to pass configuration parameters. It offers configuration flexibility. If not secured, DHCPv6 is vulnerable to various attacks, particularly spoofing attacks. This document analyzes the security issues of DHCPv6 and specifies a Secure DHCPv6 mechanism based on using CGAs.

Table of Contents

1. Introduction ................................................ 3
2. Terminology ................................................. 3
3. Security Overview of DHCPv6 ................................ 3
4. Secure DHCPv6 Overview ...................................... 4
   4.1. New Components ......................................... 5
   4.2. Support for algorithm agility .......................... 5
5. Extensions for Secure DHCPv6 ................................ 6
   5.1. CGA Parameter Option ................................... 6
   5.2. Signature Option ....................................... 7
   5.3. Signature Option for Relay-Reply Message ............... 9
   5.4. DUID-SA Type .......................................... 10
6. Processing Rules and Behaviors ............................. 11
   6.1. Processing Rules of Sender ............................ 11
   6.2. Processing Rules of Receiver .......................... 12
   6.3. Processing Rules of Relay Agent ....................... 13
7. Security Considerations .................................... 14
8. IANA Considerations ........................................ 16
9. Acknowledgments ............................................ 17
10. References ................................................ 17
   10.1. Normative References .................................... 17
   10.2. Informative References .................................. 17
1. Introduction

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6 [RFC3315]) enables DHCPv6 servers to pass configuration parameters. It offers configuration flexibility. If not secured, DHCPv6 is vulnerable to various attacks, particularly spoofing attacks.

This document analyzes the security issues of DHCPv6 in details. This document provides mechanisms for improving the security of DHCPv6:

- the address of a DHCPv6 message sender, which can be a DHCPv6 server, a relay agent or a client, can be verified by a receiver.

- The integrity of DHCPv6 messages can be checked by the receiver of the message.

The security mechanisms specified in this document is based on Cryptographically Generated Addresses (CGA [RFC3972]).

Secure DHCPv6 is applicable in environments where physical security on the link is not assured (such as over wireless) and attacks on DHCPv6 are a concern.

2. Terminology

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

3. Security Overview of DHCPv6

DHCPv6 is a client/server protocol that provides managed configuration of devices. It enables DHCPv6 server to automatically configure relevant network parameters on clients. In the basic DHCPv6 specification [RFC3315], security of DHCPv6 message can be improved in a few aspects.

a) In the basic DHCPv6 specifications, the DHCPv6 server uses a "regular" IPv6 address for itself. It is possible for a malicious attacker to use a fake address to spoof or launch an attack. See Section 23, "Security Considerations" of [RFC3315] for more details.

CGA-based security mechanism can provide proof of ownership of source addresses, which prevents such attacks.
The basic DHCPv6 specifications can optionally authenticate the origin of messages and validate the integrity of messages using an authentication option with a symmetric key pair. [RFC3315] relies on pre-established secret keys. For any kind of meaningful security, each DHCPv6 client would need to be configured with its own secret key; [RFC3315] provides no mechanism for doing this.

For the key of the hash function, there are two key management mechanisms. Firstly, the key management is out of band, usually manual, i.e. operators set up key database for both server and client before running DHCPv6. Usually multiple keys are deployed one a time and key id is used to specify which key is used.

Manual key distribution runs counter to the goal of minimizing the configuration data needed at each host. [RFC3315] provides an additional mechanism for preventing off-network timing attacks using the Reconfigure message: the Reconfigure Key authentication method. However, this method provides no message integrity or source integrity check. This key is transmitted in plaintext.

Comparing to this, the CGA-based security mechanism only require a key pair on the sender. The key management mechanism is very simple.

c) Communication between a server and a relay agent, and communication between relay agents, can be secured through the use of IPsec, as described in section 21.1 in [RFC3315]. However, IPsec is quite complicated. A simpler security mechanism, which can be easier to deploy, is desirable.

4. Secure DHCPv6 Overview

To solve the abovementioned security issues, we introduce the use of CGAs into DHCPv6. CGAs are introduced in [RFC3972]. By combining CGAs with signatures based on the CGA-associated key pair, address ownership can be verified and messages protected, "without a certification authority or any security infrastructure." [RFC3972]

This document introduces a Secure DHCPv6 mechanism that uses CGAs to secure the DHCPv6 protocol. It assumes: the secured DHCPv6 message sender already has a CGA and its corresponding CGA parameters; and the receiver has already been have the CGAs of the sender, which may be pre-configured or recorded from previous communications; in the server-relay and relay-server scenarios, the receiver has also been pre-configured the associated CGA parameters of the sender.

In this document, we introduce a CGA option with a mechanism for proving address ownership and two signature options with a corresponding verification mechanism. A DHCPv6 message (from a server, a relay agent or a client), with a CGA as source address and
carry a digital signature, can be verified by the receiver for both the CGA and signature, then process the payload of the DHCPv6 message only if the validation is successful.

This improves communication security of DHCPv6 messages. The authentication options can also be used for replay protection.

Because the sender can be a DHCPv6 server, a relay agent or a client, the end-to-end security protection can be from DHCPv6 servers to relay agents or clients, or from clients to DHCPv6 servers. Relay agents MAY add its own Secure DHCPv6 options in Relay-Forward messages when transmitting client messages to the server.

4.1. New Components

The components of the solution specified in this document are as follows:

- CGAs are used to make sure that the sender of a DHCPv6 message is the "owner" of the claimed address. A public-private key pair has been generated by a node itself before it can claim an address. A new DHCPv6 option, the CGA Parameter Option, is used to carry the public key and associated parameters.

- Signatures signed by private key protect the integrity of the DHCPv6 messages and authenticate the identity of their sender.

- Server Address type of DUID is used to carry server’s source address in the server-relay-client scenarios. The receiver gets the server’s source CGA address for CGA verification.

4.2. Support for algorithm agility

Hash functions are the fundamental security mechanism, including CGAs in this document. "...they have two security properties: to be one way and collision free." "The recent attacks have demonstrated that one of those security properties is not true." [RFC4270] It is theoretically possible to perform collision attacks against the "collision-free" property.

Following the approach recommended by [RFC4270] and [NewHash], recent analysis shows none of these attacks are currently possible, according to [RFC6273]. "The broken security property will not affect the overall security of many specific Internet protocols, the conservative security approach is to change hash algorithms." [RFC4270]

However, these attacks indicate the possibility of future real-world attacks. Therefore, we have to take into account that attacks will improved in the future, and provide a support for multiple hash
algorithms. Our mechanism, in this document, supports not only hash algorithm agility but also signature algorithm agility.

Support for hash agility within CGAs has been defined in [RFC4982]. The usage of CGAs in this document SHOULD also obey [RFC4982], too.

The support for algorithm agility in this document is mainly a unilateral notification model from a sender to a receiver. If the receiver cannot support the algorithm provided by the sender, it takes the risk itself. Senders in a same network do not have to upgrade to a new algorithm simultaneously.

5. Extensions for Secure DHCPv6

This section extends DHCPv6. Three new options and a new DUID type have been defined. The new options MUST be supported in the Secure DHCPv6 message exchange. The new DUID type MUST be supported in relay scenarios.

5.1. CGA Parameter Option

The CGA option allows the verification of the sender’s CGAs. The format of the CGA option is described as follows:

```
<table>
<thead>
<tr>
<th>option-code</th>
<th>OPTION_CGA_PARAMETER (TBA1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>option-len</td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>CGA Parameters (variable length)</td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------</td>
</tr>
</tbody>
</table>

```

option-code        OPTION_CGA_PARAMETER (TBA1).
option-len          Length of CGA Parameters in octets.
CGA Parameters      A variable-length field containing the CGA Parameters data structure described in Section 4 of [RFC3972]. This specification requires that the public key found from the CGA Parameters field in the CGA option MUST be that referred by the Key Hash field in the Signature option. Packets received with two different keys MUST be silently discarded.
5.2. Signature Option

The Signature option allows public key-based signatures to be attached to a DHCPv6 message. The Signature option could be any place within the DHCPv6 message. It protects the entire DHCPv6 header and options, particularly including the CGA option, except for the Signature option and the Authentication Option. The format of the Signature option is described as follows:

```
+-------------------+-------------------+-------------------+-------------------+
| 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 |
+-------------------+-------------------+-------------------+-------------------+
| OPTION_SIGNATURE | option-len        |
+-------------------+-------------------+-------------------+-------------------+
| HA-id             | SA-id             |
+-------------------+-------------------+-------------------+-------------------+
| HA-id-KH          | Reserved          |
+-------------------+-------------------+-------------------+-------------------+
| Timestamp (64-bit) |
+-------------------+-------------------+-------------------+-------------------+
| Key Hash (128-bit) |
+-------------------+-------------------+-------------------+-------------------+
| Signature (variable length) |
+-------------------+-------------------+-------------------+-------------------+
| Padding           |
+-------------------+-------------------+-------------------+-------------------+
```
in IANA. The initial values are assigned for RSASSA-PKCS1-v1_5 is 0x0001.

HA-id-KH
Hash Algorithm id for Key Hash. Hash algorithm used for producing the Key Hash field in the Signature option. This design is adopted in order to provide hash algorithm agility. The value is from the Hash Algorithm for Secure DHCPv6 registry in IANA. The initial values are assigned for SHA-1 is 0x0001.

Reserved
A 16-bit field reserved for future use. The value MUST be initialized to zero by the sender, and MUST be ignored by the receiver.

Timestamp
The current time of day (NTP-format timestamp [RFC5905], a 64-bit unsigned fixed-point number, in seconds relative to 0h on 1 January 1900.). It can reduce the danger of replay attacks.

Key Hash
A 128-bit field containing the most significant (leftmost) 128 bits of the hash value of the public key used for constructing the signature. The hash algorithm is indicated in the HA-id-KH field. The field is taken over the presentation used in the Public Key field of the CGA Parameters data structure carried in the CGA option. Its purpose is to associate the signature to a particular key known by the receiver. Such a key can either be stored in the certificate cache of the receiver or be received in the CGA option in the same message.

Signature
A variable-length field containing a digital signature. The signature value is computed with the hash algorithm and the signature algorithm, as described in HA-id and SA-id. The signature constructed by using the sender’s private key protects the following sequence of octets:

1. The 128-bit CGA Message Type tag value for Secure DHCPv6, 0x81be a1eb 0021 ce7e caa9 4090 0665 d2e0 02c2.
2. The 128-bit Source IPv6 Address.
3. The 128-bit Destination IPv6 Address.
4. The DHCPv6 message header.
5. All DHCPv6 options except for the Signature option and the Authentication Option.

6. The content between the option-len field and the signature field in this Signature option, in the format described above.

Padding
This variable-length field contains padding, as many bits long as remain after the end of the signature. This padding is only needed if the length of signature is not a multiple of 8 bits.

Note: a Relay-Reply message is constructed by a DHCPv6 server in segments. The server first constructs the server message for client, which includes a Signature Option that covers the server message. In the signed data, the destination address is the address of the client. It then constructs the Relay-Reply message by encapsulating the server message into a Relay Message Option. If there is additional option for relay, the server MUST include a Signature Option for Relay-Reply Message, defined below, which covers the entire Relay-Reply message. In the signed data, the destination address is the address of the target relay agent.

5.3. Signature Option for Relay-Reply Message

In the server-relay-client scenario, the Relay-Reply message may be carried two signatures: one covers the server message for client, one covers the entire relay-reply message. In order to save the double transmission of 32 byte duplicated data, which include HA-id, SA-id, SA-id-HK, Timestamp and Key Hash, another signature option is designed for Relay-Reply message only. On the receiver - the relay agent, these data can be obtained from the Signature Option within the Relay Message option. The format of the Signature Option for Relay-Reply message is described as follows:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| OPTION_SIGNATURE_RRM | option-len |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Signature (variable length) |
| Padding |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

option-code OPTION_SIGNATURE_RRM (TBA3).
option-len      Length of Signature field and Padding field in octets.

Signature       The same with Section 5.3.

Padding        The same with Section 5.3.

5.4. DUID-SA Type

Server Address Type DUID (DUID-SA) allows IPv6 address of DHCPv6 servers can be carried in DHCPv6 message payload.

The following diagram illustrates the format of a DUID-SA:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------------------------------------+
|             TBA3                         |
|------------------------------------------+  
|                                          |
+------------------------------------------+
|                                           |
|                                           |
|                                           |
+------------------------------------------+
|                                          |
|                                           |
|                                           |
|                                           |
+------------------------------------------+
|                                          |
|                                          |
|                                          |
|                                           |
+------------------------------------------+
|                                          |
|                                          |
+------------------------------------------+

Type-code       DUID-SA Type (TBA4)

Server Address   The 128-bit IPv6 address of the DHCPv6 server.

The Server Address field of DUID-SA, which is the IPv6 address of the DHCPv6 server, MUST be a CGA in the Secure DHCPv6.

In the server-relay-client scenarios, a DHCPv6 server knows a client is behind relay(s) if it receives a Relay-forward DHCPv6 message. Then it will reply a Relay-reply message. Within the payload of Relay-reply message, the server’s source CGA address is carried in the Server Address Type DUID that is encapsulated in a Relay Message option. In this way, the receiver, a DHCPv6 client can get the server’s source CGA address for CGA verification.

All the payloads, including DUID-SA, are protected by signature option by the definition of section 5.1 and 5.2.

If there is DUID-SA in the server message and its address is different from the source address of IP packet, the client MUST use the address in DUID-SA for CGA verification.
6. Processing Rules and Behaviors

6.1. Processing Rules of Sender

The sender of a Secure DHCPv6 message could be a DHCPv6 server, a DHCPv6 relay agent or a DHCPv6 client.

The node MUST have the following information in order to create Secure DHCPv6 messages:

- **CGA parameters**: Any information required to construct CGAs, as described in [RFC3972].
- **Keypair**: A public-private key pair. The public key used for constructing the signature MUST be the same in CGA parameters.
- **CGA flag**: A flag that indicates whether CGA is used or not.

To support Secure DHCPv6, the Secure DHCPv6 enabled sender MUST construct the DHCPv6 message following the rules defined in [RFC3315]. The sender MUST use a CGA, which be constructed as specified in Section 4 of [RFC3972], as the source address, unless they are sent with the unspecified source address.

A Secure DHCPv6 message MUST contain both the CGA option and the Signature option, except for the Relay-forward and Relay-reply Messages. If a relay agent adds its own options in Relay-forward message, it MUST contain the Signature option. If it does not any add new options it MUST NOT add either the CGA option or the Signature option into Relay-forward message. If a server adds addition options for relay agents in Relay-reply message, it MUST contain the Signature Option for Relay-Reply Message. If it does not add any addition options, it MUST NOT add the CGA option, the Signature option, or the Signature Option for Relay-Reply Message into the Relay-reply message.

The CGA option is constructed according to the rules presented in Section 5.1 and in [RFC3972]. The public key in the field is the one associated with the CGA, which is also the source address in the message header.

The Signature option MUST be constructed as explained in Section 5.2. It protects the message header and the message payload and all DHCPv6 options (including the CGA option) except for the Signature option itself and the Authentication Option. The Signature Option for Relay-Reply Message MUST be constructed as explained in Section 5.3.
When constructing a Relay-reply message, a DHCPv6 server MUST include an OPTION_SERVERID [RFC3315] and put its CGA in the Server Address field of the DUID in the OPTION_SERVERID in the Relay Message Option. By applying this rule, the CGA of the DHCPv6 server will not be lost when the relay agents decapsulate the Relay-reply messages, so that the client can verify CGA address and signature.

6.2. Processing Rules of Receiver

When receiving a DHCPv6 message (except for Relay-Forward and Relay-Reply messages), a Secure DHCPv6 enabled receiver SHOULD discard the DHCPv6 message if either the CGA option or the Signature option is absent. If both options are absent, the receiver MAY fallback the unsecure DHCPv6 model.

The receiving node MUST verify the source CGA address of the DHCPv6 message by using the public key of the DHCPv6 message sender, CGA Parameters and the algorithm described in Section 5 of [RFC3972]. The inputs to the algorithm are the source address, as used in IPv6 header, and the CGA Parameters field. In the relay scenarios, a DHCPv6 server obtains the CGA of a client from the peer address field in the Relay-forward message. A DHCPv6 client obtains the CGA of a server from the Server Address field of the DUID in the OPTION_SERVERID.

If the CGA verification is successful, the recipient proceeds with a more time-consuming cryptographic check of the signature. Note that even if the CGA verification succeeds, no claims about the validity of the use can be made until the signature has been checked.

The receiving node MUST verify the Signature option as follows: the Key Hash field MUST indicate the use of a known public key, the one learned from a preceding CGA option in the same message. The signature field verification MUST show that the signature has been calculated as specified in Section 5.2.

Only the messages that get through both CGA and signature verifications are accepted as secured DHCPv6 messages and continue to be handled for their contained DHCPv6 options as defined in [RFC3315]. Messages that do not pass all the above tests MUST be discarded or treated as unsecure messages.

The receiver MAY record the verified CGA for future authentications.

Furthermore, the node that supports the verification of the Secure DHCPv6 messages MAY record the following information:

Minbits The minimum acceptable key length for public keys used in the generation of CGAs. An upper limit MAY also be set for the amount of
computation needed when verifying packets that use these security associations. The appropriate lengths SHOULD be set according to the signature algorithm and also following prudent cryptographic practice. For example, minimum length 1024 and upper limit 2048 may be used for RSA [RSA].

A Relay-forward message without any addition option to Relay Message option or a Relay-forward message with both addition options and the Signature option is accepted for a Secure DHCPv6 enabled server. Otherwise, the message SHOULD be discarded or treated as unsecure message. If Signature option is presented in the Relay-forward message, the CGA verification and signature verification are needed. The server obtains the CGA parameters of the relay agents from pre-configured data. The server MUST also verify the CGA and signature for the encapsulated client DHCPv6 message in the Relay Message Option. The client CGA address is obtained from the peer-address field in the Relay-forward message.

A Relay-reply message without any addition option to Relay Message option or a Relay-reply message with both addition options and the Signature Option for Relay-Reply message is accepted for a Secure DHCPv6 enabled server. Otherwise, the message SHOULD be discarded or treated as unsecure message. If the Signature Option for Relay-Reply message is presented in the Relay-reply message, the CGA verification and signature verification are needed. The relay agents obtain the CGA parameters of the server from pre-configured data. It obtains HA-id, SA-id, SA-id-HK, Timestamp and Key Hash from Signature option encapsulated in the Relay Message option.

6.3. Processing Rules of Relay Agent

To support Secure DHCPv6, relay agents MUST follow the same processing rules defined in [RFC3315].

In the client-relay-server scenario, the relay agent MAY verify the CGA and signature as a receiver before relaying the client message further, following verification procedure define in Section 6.2. In the case of failure, it MUST discard the DHCPv6 message. However, this does not save the load of the DHCPv6 server. The server still MUST verify the CGA and signature by itself in order to prevent the attack between the relay agent and server.

In the server-relay-client scenario, if the Signature Option for Relay-Reply message is presented, the relay agent MUST verify the CGA and signature before relaying the server message further, following verification procedure define in Section 6.2. In the case of failure, it MUST discard the DHCPv6 message.
In the relay scenarios, because relay agents restructure the DHCPv6 messages, a downstream receiver would not find the sender’s source CGA address in the DHCPv6 message header.

In the client-relay-server scenarios, "The relay agent copies the source address from the IP datagram in which the message was received from the client into the peer-address field in the Relay-forward message" [RFC3315]. Therefore, the CGA of a client will not be lost during the relay processing from the client to the server. The receiver, a DHCPv6 server, can find the sender’s source CGA address in the peer-address field for CGA verification.

During the relay processing from the server to the client, when the relay agent constructs the IPv6 header for the server message, the source IPv6 address is the relay’s IPv6 address, rather than the server’s IPv6 address. In order to make the CGA of the DHCPv6 server reach the client, DUID-SA, described in Section 5.4, MUST be used. Defined in [RFC6422], "the implicit requirement that relay agents not modify the content of encapsulation payloads as they are relayed back toward clients", A relay agent will not change the OPTION_SERVERID when processing Relay-reply message from a DHCPv6 server, so that the CGA of the DHCPv6 server will not be lost when the Relay-reply message is decapsulated in the relay agent. The relay agent MAY also verify the CGA and signature for the encapsulated DHCPv6 message in the Relay Message Option. This can be helpful if the DHCPv6 response traverses a separate administrative domain, or if the relay agent is in a separate administrative domain. However, this is not necessary because the DHCPv6 client validation will catch any modification to the response.

7. Security Considerations

This document provides new security features to the DHCPv6 protocol.

Using CGA as source addresses with its verification mechanism in DHCPv6 message exchanging provides the source address ownership verification and data integrity protection.

The Secure DHCPv6 mechanism is based on the pre-condition that the receiver knows the CGA of senders. For example, to prevent DHCPv6 server spoofing, the clients should be pre-configured with the DHCPv6 server CGA. The clients may decline the DHCPv6 messages from unknown servers, which may be fake servers; or may prefer DHCPv6 messages from known servers over unsigned messages or messages from unknown servers. The pre-configuration operation also needs to be protected, which is out of scope.

In the relay-server and server-relay authentication scenarios, the Secure DHCPv6 mechanism is based on the pre-condition that the receiver has been pre-configured with sender’s CGAs and associated
CGA parameters. The pre-configuration operation also needs to be protected, which is out of scope.

CGA-based signatures cannot be used to authenticate a transaction if the CGA key isn’t pre-configured in the DHCPv6 client that needs to authenticate the transaction. However, such a DHCPv6 client can make a leap of faith when it first encounters a new CGA. If the DHCPv6 server that used that CGA is in fact legitimate, then all future communication with that DHCPv6 server can be protected by caching the CGA and the associated public key. This does not provide complete security, but it limits the opportunity to mount an attack on a specific DHCPv6 client to the first time it communicates with a new DHCPv6 server.

DHCPv6 nodes without CGAs or the DHCPv6 messages that use unspecific addresses cannot be protected.

Downgrade attacks cannot be avoided if nodes are configured to accept both secured and unsecured messages. A future specification may provide a mechanism on how to treat unsecured DHCPv6 messages.

As stated in CGA definition [RFC3972], link-local CGAs are more vulnerable because the same prefix is used by all IPv6 nodes. Therefore, when link-local CGAs are used by the DHCPv6 clients, it is recommended to use a slightly higher Sec value, for example Sec=1 for now. When higher Sec values are used, the relative advantage of attacking link-local addresses becomes insignificant.

Impacts of collision attacks on current uses of CGAs are analyzed in [RFC4982]. The basic idea behind collision attacks, as described in Section 4 of [RFC4270], is on the non-repudiation feature of hash algorithms. However, CGAs do not provide non-repudiation features. Therefore, as [RFC4982] points out CGA-based protocols, including Secure DHCPv6 defined in this document, are not affected by collision attacks on hash functions.

[RFC6273] has analyzed possible threats to the hash algorithms used in SEND. Since the Secure DHCPv6 defined in this document uses the same hash algorithms in similar way to SEND (except that Secure DHCPv6 has not used PKIX Certificate), analysis results could be applied as well: current attacks on hash functions do not constitute any practical threat to the digital signatures used in the signature algorithm in the Secure DHCPv6. Attacks on CGAs, as described in [RFC4982], will compromise the security of Secure DHCPv6 and they need to be addressed by encoding the hash algorithm information into the CGA as specified in [RFC4982].
8. IANA Considerations

This document defines two new DHCPv6 [RFC3315] options, which MUST be assigned Option Type values within the option numbering space for DHCPv6 messages:

   The CGA Parameter Option (TBA1), described in Section 5.1.
   The Signature Option (TBA2), described in Section 5.2.
   The Signature Option for Relay-Reply Message (TBA3), described in Section 5.3.

This document defines a new DHCPv6 DUID, which MUST be assigned DUID Type values within the DHCPv6 DUID Type numbering space:

   The DUID-SA (TBA4), described in Section 5.4.

This document defines two new registries that have been created and are maintained by IANA. Initial values for these registries are given below. Future assignments are to be made through Standards Action [RFC5226]. Assignments for each registry consist of a name, a value and a RFC number where the registry is defined.

Hash Algorithm for Secure DHCPv6. The values in this name space are 16-bit unsigned integers. The following initial values are assigned for Hash Algorithm for Secure DHCPv6 in this document:

   Name   |  Value  |  RFCs
   -----------------------+---------+------------
   Reserved             |  0x0000 | this document
   SHA-1                |  0x0001 | this document
   SHA-256              |  0x0002 | this document

Signature Algorithm for Secure DHCPv6. The values in this name space are 16-bit unsigned integers. The following initial values are assigned for Signature Algorithm for Secure DHCPv6 in this document:

   Name   |  Value  |  RFCs
   -----------------------+---------+------------
   Reserved             |  0x0000 | this document
   RSASSA-PKCS1-v1_5    |  0x0001 | this document

This document defines a new 128-bit value under the CGA Message Type [RFC3972] namespace, 0x81be a1eb 0021 ce7e caa9 4090 0665 d2e0 02c2. (The tag value has been generated randomly by the editor of this specification. It may replaced by any IANA-allocated value when the specification is published.)
9. Acknowledgments

The authors would like to thank Bernie Volz, Ted Lemon, Ralph Dorms, Jari Arkko, Sean Turner, Stephen Kent, Thomas Huth, David Schumacher and other members of the IETF DHC and CSI working groups for their valuable comments.

10. References

10.1. Normative References


10.2. Informative References


Author’s Addresses

Sheng Jiang
Huawei Technologies Co., Ltd
Q14, Huawei Campus
No.156 Beiqing Road
Hai-Dian District, Beijing 100095
P.R. China
EMail: jiangsheng@huawei.com

Sean Shen
CNNIC
4, South 4th Street, Zhongguancun
Beijing 100190
P.R. China
EMail: shenshuo@cnnic.cn
A Generic IPv6 Addresses Registration Solution
Using DHCPv6
draft-jiang-dhc-addr-registration-04.txt

Status of this Memo

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

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

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

This Internet-Draft will expire on August 29, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
Abstract

In the IPv6 address allocation scenarios, host self-generated addresses are notionally conflicted with the network managed address architecture. These addresses need to be registered in the networking management plate for the purposes of central address administration. This document introduces a generic address registration solution using DHCPv6, and defines one new ND option and one new DHCPv6 option in order to propagate the solicitations of registering self-generated addresses. The registration procedure reuses the existing IA_NA in the DHCPv6 protocol.

Table of Contents

1. Introduction & Requirements .................................. 3
2. Terminology .................................................. 3
3. Overview of Generic Address Registration Solution .......... 3
4. Propagating the Address Registration Solicitation .......... 4
   4.1. ND Address Registration Solicitation Option ............. 5
   4.2. DHCPv6 Address Registration Solicitation Option ........ 6
5. DHCPv6 Address Registration Procedure ........................ 7
   5.1. DHCPv6 Address Registration Request ..................... 7
   5.2. DHCPv6 Address Registration Acknowledge ................ 8
6. Security Considerations ...................................... 8
7. IANA Considerations .......................................... 8
8. Acknowledgments .............................................. 9
9. References ................................................... 9
   9.1. Normative References .................................... 9
   9.2. Informative References .................................. 10
Author’s Addresses ............................................. 10
1. Introduction & Requirements

In the IPv6 address allocation scenarios, there are many host self-generated addresses, such as addresses in IPv6 Stateless Address Configuration [RFC4862, RFC4941] scenario and Cryptographically Generated Addresses (CGA, [RFC3972]), and etc. These addresses are notionally conflicted with the network managed address architecture, such as Dynamic Host Configuration Protocol for IPv6 (DHCPv6, [RFC3315]) managed network or network with Access Control List.

Many operators of enterprise networks and similarly tightly administered networks have expressed the desire to be at least aware the hosts’ addresses when moving to IPv6. Furthermore, they may want to stop the usage of some hosts’ addresses for various reasons.

A useful way to give network administrators most of what they want, while at the same time retaining compatibility with normal stateless configuration would be: if the self-generated IPv6 addresses are used, they may need to be registered in the networking management plate. The host may be required to perform this registration since only registered IPv6 addresses may access the network resources in some scenarios.

In order to fulfill the abovementioned practice, this document introduces a new Neighbor Discovery (ND) option and a new DHCPv6 option to propagate the address registration solicitation from network management to hosts. DHCPv6 protocol is suitable to perform the address registration procedure while the address registration server may play by a DHCPv6 server or a stand-alone server. The existing IA_NA in the DHCPv6 protocol is reused for the registration procedure.

2. Terminology

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

3. Overview of Generic Address Registration Solution

By current default, the hosts with self-generated addresses do not register their addresses to any network devices. However, this may result that the network may reject the access request from these devices if the address registration is requested.
As showed in below Figure 1, in the generic address registration solution, proposed by this document, the network management plate firstly propagates the solicitations of registering self-generated addresses, by messages from either local router (step 1a in Figure 1) or DHCPv6 server (step 1b in Figure 1).

By received such solicitations, a host using the self-generated address SHOULD send an address registration request message to the address registration server (step 2 in Figure 1). The address registration server may be acted by a DHCPv6 server. By received the address registration request, the address registration server records the requested address in the address database, which MAY be used by other network functions, such as DNS or ACL, etc. The address registration server should also assign lifetimes for the requested address. An acknowledgement is sent back to the host with the assigned lifetimes (step 3 in Figure 1).

By received the acknowledgement, the host can use the registered address. It SHOULD use the assigned preferred and valid lifetime for the corresponded address.

4. Propagating the Address Registration Solicitation

In order to indicate or force the hosts with self-generated addresses to register their addresses and the appointed address registration server, new solicitation options need to be defined.

There are more than one mechanisms in which configuration parameters could be pushed to the end hosts. The address registration
solicitation option can be carried in Router Advertisement (RA) message, which is broadcasted by local routers. In the DHCPv6 managed network, it can also be carried in DHCPv6 messages.

More precisely it defines one new ND option and one new DHCPv6 option that convey a Fully Qualified Domain Name (FQDN, as per Section 3.1 of [RFC1035]) of address registration(s). In order to make use of these options, this document assumes appropriate name resolution means (see Section 6.1.1 of [RFC1123]) are available on the host client. The use of the FQDN may benefit for load-balancing purposes.

By receiving the address registration solicitation option(s), a host SHOULD register its self-generated addresses, if there are any, to the appointed registration server. The solicitation options may include the IPv6 address(es) of address registration server.

In principle, hosts must receive a prefix from either RA message [RFC4861] or DHCPv6 message [I-D.ietf-dhc-host-gen-id] so that they can generate an IPv6 address by themselves. The Address Registration Solicitation options could be propagated together with prefix assignment information.

4.1. ND Address Registration Solicitation Option

The ND Address Registration Solicitation Option allows routers to propagate the solicitation for hosts to register their self-generated address. This option also carries a domain name of the appointed address registration server. This option SHOULD be propagated together with ND Prefix Information Option, Section 4.6.2, [RFC4861]. The format of the ND Address Registration Solicitation Option is described as follows:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |  Pad Length   |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                          Domain Name                          .
|. (Address Registration Server)                                .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                            Padding                            .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
Fields:

Type      (TBA1)

Length     The length of the option in units of 8 octets, including the Type and Length fields. The value 0 is invalid. The receiver MUST discard a message that contains this value.

Pad Length The number of padding octets beyond the end of the Domain Name field but within the length specified by the Length field.

Reserved   Padding bits. It is for future use also. The value MUST be initialized to zero by the sender, and MUST be ignored by the receiver.

Domain Name A fully qualified domain name of the appointed address registration server. The domain name is encoded as specified in Section 8 of [RFC3315]. Any possible future updates to Section 8 of the Section 8 of [RFC3315] also apply to this option.

Padding: A variable-length field making the option length a multiple of 8, containing as many octets as specified in the Pad Length field. Padding octets MUST be set to zero by senders and ignored by receivers.

4.2. DHCPv6 Address Registration Solicitation Option

The DHCPv6 Address Registration Solicitation Option allows DHCPv6 server to propagate the solicitation for hosts to register their self-generated address. This option also carries a domain name of the appointed address registration server. This option SHOULD be propagated together with DHCPv6 Prefix Information Option, Section 5, [I-D.ietf-dhc-host-gen-id]. The format of the DHCPv6 Address Registration Solicitation Option is described as follows:
5. DHCPv6 Address Registration Procedure

The current DHCPv6 protocol is reused as the address registration protocol while a DHCPv6 server plays as address registration server. Identity Association for Non-temporary Addresses (IA_NA) [RFC3315] is reused in order to fulfill the address registration interactions.

5.1. DHCPv6 Address Registration Request

The host with self-generated address(es) sends a DHCPv6 Request message to the appointed address registration server, which may be a DHCPv6 server.

The DHCPv6 Request message SHOULD contain at least one IA_NA option. The IA_NA option SHOULD contain at least one IA Address option. The host SHOULD set the T1 and T2 fields in any IA_NA options, and the preferred-lifetime and valid-lifetime fields in the IA Address options to 0.

By received, the address registration server MUST register the requested address in its address database, which MAY be used by other network functions, such as DNS or ACL, etc. The address registration server SHOULD also assign the lifetimes for these registered addresses.
5.2. DHCPv6 Address Registration Acknowledge

The address registration server sends a Reply message as the response to registration requests.

The DHCPv6 Reply message SHOULD contain at least one IA_NA option. The IA_NA option SHOULD contain at least one IA Address option. The server SHOULD set the T1 and T2 fields in any IA_NA options, and the preferred-lifetime and valid-lifetime fields in the IA Address options following the rules defined in Section 22 in [RFC3315].

By received the acknowledgement from the server, the host can use the registered address to access the network. It SHOULD use the values in the preferred and valid lifetime fields for the preferred and valid lifetimes of the address.

Note: the host MAY continue to use expired address, such as Locators as Upper-Layer Identifiers (ULID) in Shim6 protocol [RFC5533], etc.; but the network MAY refuse the network access from such addresses.

6. Security Considerations

An attacker may use a faked address registration request option to indicate hosts reports their address to a malicious server and collect the user information. These attacks may be prevented by using Secure Neighbor Discovery (SEND, [RFC3971]) if RA Address Registration Request Option is used, or AUTH option [RFC3315] or Secure DHCPv6 [I-D.ietf-dhc-secure-dhcpv6] if DHCPv6 Address Registration Request Option is used.

7. IANA Considerations

This document defines a new Neighbor Discovery [RFC4861] option, which MUST be assigned Option Type values within the option numbering space for Neighbor Discovery Option Type:

The Address Registration Solicitation Option (TBA1), described in Section 4.1.

This document defines one new DHCPv6 [RFC3315] option, which MUST be assigned Option Type values within the option numbering space for DHCPv6 options:
The OPTION_Addr_Reg_Solicitation (TBA2), described in Section 4.2;

8. Acknowledgments

The authors would like to thank Ralph Dorm, Ted Lemon, Bernie Volz and other member of DHC WG for valuable comments.

9. References

9.1. Normative References


9.2. Informative References

[I-D.ietf-dhc-secure-dhc-pv6]

[I-D.ietf-dhc-host-gen-id]

Author’s Addresses

Sheng Jiang
Huawei Technologies Co., Ltd
Q14, Huawei Campus
No.156 Beiqing Road
Hai-Dian District, Beijing 100095
Phone: 86-10-82882681
Email: jiangsheng@huawei.com

Gang Chen
China Mobile
53A, Xibianmennei Ave., Xuanwu District, Beijing
P.R. China
Phone: 86-13910710674
Email: phdgang@gmail.com
Populating the DNS Reverse Tree for DHCP Delegated Prefixes
draft-lemon-dhc-dns-pd-00.txt

Abstract

This document describes three alternatives for populating the DNS reverse tree for prefixes delegated using DHCP, and provides mechanisms for implementing each alternative.

Status of this Memo

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

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

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

This Internet-Draft will expire on September 7, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
Table of Contents

1. Introduction ................................................. 3
2. Terminology .................................................. 4
3. Methods for populating the reverse tree ...................... 5
   3.1. Site-managed reverse tree ................................. 5
   3.2. Provider-managed reverse tree ............................ 5
   3.3. Provider-managed spoofed reverse tree ..................... 5
   3.4. Other solutions not documented here ..................... 5
4. Negotiating the reverse tree population method ............... 7
5. Configuring a site-managed reverse tree ...................... 9
   5.1. Requesting Router Behavior .............................. 9
   5.2. Delegating Router Behavior ............................... 10
6. Configuring a provider-managed reverse tree .................. 12
   6.1. Requesting Router Behavior .............................. 12
   6.2. Delegating Router Behavior ............................... 12
7. Configuring a spoofed reverse tree ............................ 13
8. Configuring no reverse tree .................................. 14
9. Security Considerations ...................................... 15
10. IANA Considerations ......................................... 16
11. Normative References ....................................... 17
Author’s Address ................................................ 18
1. Introduction

When a site is numbered using DHCP prefix delegation [RFC3633], there are three ways of populating the Domain Name System [RFC1035] reverse tree. Which mechanism is chosen depends on the capabilities of the site’s DNS infrastructure, if any, on the capabilities and policies of the service provider, and on the preferences of the site administration.

This document does not take a position on which mechanism, if any, is best for populating the reverse tree, but simply documents each of the possible mechanisms for doing so, and provides a means whereby site administrators and service providers can negotiate the mechanism whereby the reverse tree for a particular site will be populated.
2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].
3. Methods for populating the reverse tree

There are three common methods of populating the reverse tree for a delegated prefix: delegation, dynamic dns, and zone spoofing. In addition, of course, it is possible to leave the reverse tree unpopulated.

3.1. Site-managed reverse tree

To populate the reverse tree by delegation, the site administrator must provide a DNS authoritative name server for the delegated zone. The site administrator must communicate the IP address of the authoritative name server to the service provider. The service provider must then add a delegation for that zone using the IP address or addresses of the DNS authoritative servers provided by the site administrator.

3.2. Provider-managed reverse tree

To populate the reverse tree using DNS updates, the service provider must provide an authoritative name server for the zone. The site administrator must provide a key to the service provider that can be used to authenticate DNS updates. The site administrator must then provide a mechanism whereby DNS updates will automatically be generated, using the provided key, whenever IP addresses are allocated within the delegated prefix.

3.3. Provider-managed spoofed reverse tree

In some cases the site administrator may not be willing or able to populate a reverse tree. However, the service provider may wish to provide meaningful answers to reverse zone queries for the delegated zone. It’s not possible to populate the delegated zone: a fully populated zone for a /64 would require 1.8x10^19 names. However, the names in such a zone would never change; consequently it is possible for a name server to spoof the zone contents, constructing answers for queries against any name within the zone on the fly. Because the contents of the zone never change, the zone can have a consistent authority record.

3.4. Other solutions not documented here

It’s worth noting that there are several other ways that the zone for a delegated prefix could be populated, but we are not covering these mechanisms because they seem more difficult to implement and deploy. For instance, nodes configured with addresses within a delegated prefix could issue their own DNS updates to an authoritative server operated by the service provider. The problem of key management in
this case becomes intractable, however.

It would also be possible for the site to have its own key management infrastructure, and for some agent on the requesting router to act as an intermediary in updating a zone maintained by the service provider. However, this is substantially more complicated than either of the proposed solutions.

Another option is to simply not populate the reverse tree. This is an attractive option in the IETF in particular because the reverse tree is frequently used for purposes to which it is not suited, and some IETF participants believe that in order to discourage these applications, it’s better simply to not populate the reverse tree. This document takes no position on this question, but does offer a means whereby the site administrator can indicate that the reverse tree should not be populated.
4. Negotiating the reverse tree population method

The prefix delegation process is initiated by a requesting router. If a delegating router chooses to delegate a prefix to the requesting router, it replies with a prefix. The requesting router may receive responses from more than one delegating router, and may choose one or more such delegated prefixes. For delegating routers whose offer is accepted, the requesting router sends a request for the offered address; at this point the delegating router commits the delegation to stable storage and sends a confirmation to the requesting router.

The messages used to complete this transaction are the DHCP Discover, DHCP Advertise, DHCP Request and DHCP Reply messages, respectively. The negotiation as to how the reverse tree will be populated piggybacks on this four-message process.

In the DHCP Discover message, the requesting router indicates the site administrator’s preference for how the reverse tree for the delegated prefix will be populated. It does this by including, in each IA_PD option it sends, a Prefix Delegation Zone Preference option (PDZP) containing one or more preference codes. These codes are listed in order of preference with the most preferred mechanism first. A requesting router that includes a PDZP option MUST send an Option Request option (ORO) that requests the Prefix Delegation Zone Method (PDZM) option.

If the delegating router chooses not to delegate a prefix to the requesting router, no special action need be taken in response to the PDZP option. The remainder of this section describes what happens if the delegating router chooses to delegate a prefix to the requesting router.

Delegating routers that implement this specification can be configured with a list of supported reverse tree population methods. When a requesting router receives an IA_PD option that includes a PDZP option, if it has been configured with a reverse tree population method list, it iterates across the list of methods in the PDZP option. For each entry in the PDZP option, the requesting router tests to see if that method has been configured by the site administrator as being supported. If the method is on the list, the iteration stops at this point.

Upon completion of this iteration, if a method was found in the PDZP that is supported by the delegating router, that is the method that will be used to populate the reverse tree for the delegated zone. The delegating router constructs a PDZM option indicating that this method will be used and includes this in the DHCP Advertise message.
If no supported method was found, this means that the service provider will not cooperate with the site administrator in populating the reverse tree. The delegating router indicates that this is the case by not including a PDZM option in the DHCP Advertise message.

The requesting router may receive one or more DHCP Advertise messages containing delegated prefixes. The requesting router MUST silently discard any DHCP Advertise message containing a PDZM option that indicates a method that was not listed in the PDZP option sent in the DHCP Discover message.

The requesting router may then choose to respond to one or more of the remaining DHCP Advertise messages, if any. The lack of a PDZM option indicates either that the delegating router does not implement DNS for delegated prefixes, or that it is not configured to support DNS for delegated prefixes. The requesting router MAY prefer DHCP Advertise messages containing PDZM options over DHCP Advertise messages that do not contain PDZM options.

When responding to any DHCP Advertise messages containing PDZM options, the requesting router MUST include a PDZM option containing the same method indicated in the received PDZM option.

Each delegating router that receives a DHCP Request message containing a PDZM option MUST check the method indicated in the PDZM option is supported; if not, the delegating router MUST silently discard the DHCP Request option.

The requesting and delegating routers should follow the same procedure specified for the DHCP Request/DHCP Reply sequence whenever a DHCP Renew or DHCP Rebind is sent and a DHCP reply sent in response, if that response renews the delegated prefix. In the case that the response does not renew the prefix, the delegating router MUST NOT send a PDZM in the IA_PD option.
5. Configuring a site-managed reverse tree

If the PDZM option returned by the delegating router in the DHCP Advertise message specifies the Site Managed method, the requesting router must arrange to set up one or more authoritative name servers that will provide service for the zone or zones that correspond to the delegated prefix. It must also communicate to the delegating router the IP address or addresses of these servers.

5.1. Requesting Router Behavior

The requesting router MUST include a Prefix Delegation Zone Server (PDZS) option in each IA_PD in the DHCP Request message, which includes zero or more IP addresses of authoritative name servers for the delegated zone. IPv4 addresses MUST be represented as IPv4-Embedded IPv6 addresses using the Well-Known prefix [RFC6052].

Authoritative name service for these zones may be provided by any or all of the following three types of authoritative name servers:

- An authoritative name server running on a node that has an IP address known to the requesting router that is not obtained from the prefix being delegated.
- An authoritative name server running on the requesting router.
- An authoritative name server running on a node that will obtain its only IP address from the prefix being delegated.

In the first case, it is possible that the reverse zone for the delegated prefix is already configured on the authoritative name server. In this case, the requesting router SHOULD include the IP address of the delegated zone in the PDZS option.

However, if the prefix is being delegated for the first time, the delegating router will not have had an opportunity to configure it prior to sending the DHCP Request message. In this case, the delegating router SHOULD NOT include the IP Address of this name server in the PDZS option that’s send in the DHCP Request message; instead, it should send a DHCP Renew once the authoritative server has been configured, and list the server’s IP address in the PDZS option in the DHCP Renew message.

In the second case, the requesting router may already have an IP address, and may be able to configure the authoritative server for the delegated zone before sending the DHCP Request. In this case, the requesting router SHOULD include its own IP address in the PDZS option in the DHCP Request message.
If the requesting router does not have an IP address at this time, it SHOULD send a DHCP Renew message containing a PDZS option that lists all the authoritative servers for the reverse zone or zones for the delegated prefix after it has an IP address and has configured the authoritative servers.

If the authoritative name server is running on a node that will configure its IP address from the delegated prefix, this name server cannot even be configured until it has an IP address. The process of configuring this name server is beyond the scope of the document; however, once the name server has been configured, the requesting router SHOULD send a DHCP Renew message for the delegated prefix with an IA_PD containing a PDZS option that lists the IP address of this name server.

In general, if there are any globally-reachable name servers that are authoritative for the zone or zones that provide the reverse tree for the delegated prefix at the time that the DHCP Request message is sent, the requesting router should list the IP addresses of these name servers in the PDZS option in the associated IA_PD option in the DHCP Request message.

If new globally-reachable name servers that are authoritative for the reverse zone or zones become available after the DHCP Request has been sent and the DHCP Reply received, the requesting router SHOULD send a DHCP Renew message containing an IA_PD for the delegated prefix and a PDZS option listing the name servers for that prefix that have come online. The requesting router SHOULD be aware of all outstanding name server configuration processes and minimize the number of DHCP Renew message sent.

When a requesting router sends a DHCP Renew or DHCP Rebind message to renew a delegated prefix, if a site-managed reverse tree was successfully configured, the requesting router MUST send a PDZM option containing the same method sent in the original DHCP Request message. The requesting router MUST also send a PDZS option that contains one or more IP addresses for authoritative servers for the reverse tree for the delegated prefix.

5.2. Delegating Router Behavior

When a delegating router receives a valid DHCP Request message containing an IA_PD that contains both a PDZM option indicating the Site Managed method and a PDZS option containing at least one IP address, it compares the IP addresses in the PDZM option to any previous record it may have for that delegation. If the contents of the PDZM option differ from the previous record, or if there is no previous record, the delegating router MUST issue a DNS Update to add
a delegation to the parent zone of the reverse tree zone for the
delegated prefix.

In the event that the PDZS option contains zero IP addresses, the
deleagating router does not update the zone.

If the delegated prefix must be represented as more than one zone,
the delegating router adds delegations to the parent zone for each
such zone.

When a delegating router receives a DHCP Renew or DHCP Rebind message
for a prefix it delegated and elects to renew the prefix, it MUST
check its record for that prefix to see if a delegation exists. If
the contents of the PDZS differ from the recorded list of
authoritative name servers for that prefix, the delegating router
MUST update the parent zone with the new delegations.

When a delegating router receives a DHCP Renew or DHCP Rebind message
for a prefix it delegated, and elects not to renew the delegation,
the delegating router MUST check to see if it has a site-managed
reverse tree configuration for that prefix. If it does, it must
update the parent zone to remove any delegations that were added, and
update its record for the delegated prefix to indicate that no site-
managed reverse tree configuration for that prefix is present.

When a delegated prefix expires without being renewed by the
requesting router, the same procedure should be followed to update
the parent zone.
6. Configuring a provider-managed reverse tree

If the PDZM returned by a delegating router in the DHCP Advertise message specifies the Provider Managed method, the delegating router must arrange to set up a reverse zone for the delegated prefix. The requesting router must communicate a key to the delegating router that can be used to secure updates to the reverse zone.

6.1. Requesting Router Behavior

In order to update the provider-managed reverse zone, the requesting router must provide a key to the delegating router. Because DHCP does not provide confidentiality, this key must be the public half of a private key.

Typically sites that wish to populate their reverse tree with meaningful information maintain a site-specific or company-wide DNS zone. In order to update the reverse zone, the site administrator must publish a SIG(0) key in this zone. The requesting router MUST include a Prefix Delegation SIG(0) Key FQDN (PDSKF) option in the DHCP Request message and any subsequent DHCP Renew messages. It must use the private half of the SIG(0) key in any DNS updates to the reverse zone.

6.2. Delegating Router Behavior

There are two cases that the delegating router needs to handle: the case where the prefix being delegated was previously delegated to the same requesting router, and the case where it was not.

In the case where the prefix was previously delegated to the same requesting router, the delegating router need take no action to populate the zone, because it should already be populated.

In the case where the prefix was previously delegated to a different requesting router, the delegating router MUST remove the old zone information from the master authoritative name server for the zone.

In this case, and in the case where no previous delegation had been done, the delegating router must then configure a new reverse zone on the master server.

In any case, the delegating router must configure the reverse zone so that it can be updated using the SIG(0) key stored on the name provided by the requesting router in the PDSKF option.
7. Configuring a spoofed reverse tree

A spoofed reverse tree can be configured either unilaterally by the service provider or upon request of the site administrator. The site administrator would list this as an option to indicate a preference for a spoofed reverse tree over no reverse tree; the choice doesn’t make any sense otherwise.

Generally speaking, the service provider has the option of either setting up spoofed zones on demand, or setting them up when requested. If the service provider only offers spoofed zones, it makes some sense to set them up in advance; otherwise they should be set up whenever a prefix is delegated to a particular requesting router for the first time.

In some cases the site administration may request a spoofed zone because they do not wish to populate the reverse tree, but wish for it to appear populated. A service provider may support this option in addition to the site-managed option, the provider-managed option, and the no zone option. In this case, when a prefix is delegated to a new router for the first time, there may be an old zone configured differently. In this case, the delegated router MUST remove the old zone configuration before setting up the spoofed zone.
8. Configuring no reverse tree

A service provider may choose to simply not populate reverse trees for delegated prefixes. This is a desirable option in the sense that it minimizes the work required to support the reverse DNS tree, and avoids creating spoofed nonsense records. The service provider may also simply offer it as an option for sites that prefer not to have a populated reverse tree.

In this case, if the non-populated reverse tree is an option, and the prefix had previously been delegated to a different router, the delegating router must remove any previously-existing zone for the delegated prefix.
9. Security Considerations
10. IANA Considerations
11. Normative References


Author’s Address

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

Phone: +1 650 381 6000
Email: mellon@nominum.com
DHCP Options for 3GPP Service
draft-liu-dhc-3gpp-option-01.txt

Abstract

This document defines a new option that can be used by DHCP clients in their signaling sent to DHCP servers, when those clients need to associate a request for an IP address or IPv6 prefix with a given 3GPP packet service. It is intended for scenarios of interworking between a non-3GPP access and a 3GPP network.

Status of this Memo

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

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

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

This Internet-Draft will expire on September 13, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents.
carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction ....................................................... 3
2. Convention & Terminology ...................................... 4
3. Problem Statement ............................................. 5
4. 3GPP-Service Option Format ................................... 6
   4.1. APN Sub-option Format .................................... 6
   4.2. Non Seamless WLAN Offload
        (3GPP-Service-Type)Sub-option Format .................. 7
5. DHCP Client Behavior ........................................... 8
6. DHCP Server Behavior ........................................... 9
7. Security Considerations ....................................... 10
8. IANA Considerations .......................................... 11
9. Normative References .......................................... 12
Authors’ Addresses .............................................. 13
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 defines the use of 3GPP specific option transferred to DHCP server by DHCP client.
2. Convention & Terminology

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

The terminology in this document is based on the definitions in [RFC2131][RFC3315], in addition to the ones specified in this section.

3GPP 3rd Generation Partnership Project
TS Technical Specification
EPC Evolved Packet Core
APN Access Point Name
PDN Packet Data Network
GTP GPRS Tunnelling Protocol
PMIP Proxy Mobile IP
3. Problem Statement

In 3GPP TS 23.402, 3GPP UE can access the 3GPP EPC through non-3GPP IP access.

When the non 3GPP access is Trusted, there is no need for the 3GPP UE to establish a Layer 3 tunnel (IPSec [RFC4301], DSMIPv6 [RFC 5555]) to access the EPC as it can rely upon the non 3GPP access security mechanisms. In this case, the 3GPP UE may send DHCP signaling to non 3GPP access to acquire an IP address. However, the 3GPP UE may wish to associate its request for an IP address with IP services provided by the 3GPP EPC or may conversely explicitly require that its traffic is not sent to the EPC but offloaded, even though the UE has been authenticated via its 3GPP credentials (the service is called "NSO" i.e. Non Seamless Offload).

The APN (Access Point Name) is the parameter by which a 3GPP UE signals the kind of EPC service it desires. Based on the value of this parameter, a 3GPP IP Edge (a PDN GW) is selected and the PDN GW is able to determine the IP features to deliver to the traffic exchanged by the UE as part of this APN.

It should be noted that the set of IP configuration parameters that the UE may receive via a DHCP response (e.g. the DNS server(s) address) may also be influenced by the value of the APN.

Thus when requesting an IP address via DHCP, a 3GPP UE should be able to indicate:

- Whether this IP address is for NSO or whether it is for an EPC service.
- If the IP address is for an EPC service, the value of the APN corresponding to the IP services reached when using this IP address. The UE may also request an EPC service without providing an APN value; in this case, the network uses a default APN value.

Such DHCP request may be sent when the UE requests access to EPC via non 3GPP access, or when the UE performs mobility from a 3GPP access to a non 3GPP access.

This document is intended to define a new DHCP option for "3GPP Service".
4.  3GPP-Service Option Format

A DHCP client within a 3GPP based device sets the "3GPP-Service" Option in the DHCP request it sends to the network to indicate the desired 3GPP service associated with that request.

A DHCP server willing to indicate it has taken into account the parameters / sub-options of a "3GPP-Service" Option included by the client in a DHCP request mirrors these parameters in the DHCP signaling it sends back to the client.

0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  3GPP-SERVICE | option-length |   sub-options...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

option-code       3GPP-Service (TBD)
option-len        The number of octets in the option, minimum 1.
Sub-options       The "3GPP-Service" Option may contain one or more Sub-options further defined in this document

Figure 1: 3GPP Service Option Format

4.1.  APN Sub-option Format

The purpose of 3GPP-APN (Access Point Name) Sub-option is sent by UE to network for identifying the packet data network associated with the DHCP message. The APN is defined in 3GPP TS 23.401[TS23401].

0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |    Length     |  Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Code          3GPP-APN(TBD).
Length        The number of octets in the option, minimum 1.
Value         The APN value, as defined in 3GPP TS 24.008 [TS24008] section 0.5.6.1.

Figure 2: APN Sub-option Format

The 3GPP-APN Sub-option SHOULD only be present once in a DHCP message. If it is present more than once, the value of the last occurrence of the option supersedes the value of the other occurrences of this sub-option.
4.2. Non Seamless WLAN Offload (3GPP-Service-Type) Sub-option Format

The purpose of 3GPP-Service-Type (Non Seamless Offload) Sub-option is to identify whether the DHCP message is to be associated with a Non Seamless Offload service or with an EPC service.

The 3GPP Non Seamless Offload service is defined in 3GPP TS 23.402 [TS23402].

```
0 1 2 3 4 5 6 7 8 9
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Code  |S| V |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- **Code**: 3GPP-Service-Type (TBD).
- **S**: Bit 4 of octet 1 is spare and shall be coded as zero.
- **V**: The 3GPP-Service-Type value, the definition is as follows,
  
<table>
<thead>
<tr>
<th>Bits</th>
<th>NSO</th>
<th>EPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>All other values are reserved.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The presence of the 3GPP-Service-Type Sub-option in a DHCP message indicates whether the DHCP signalling is associated with a NSO service or an EPC service rendered to the UE.

The 3GPP-Service-Type Sub-option SHOULD NOT be provided in the same DHCP message than the 3GPP-APN Sub-option, when the 3GPP-Service-Type sub-option includes NSO indication. If present, the 3GPP-APN Sub-option SHOULD be provided in the same DHCP message than the 3GPP-Service-Type Sub-option, when the 3GPP-Service-Type sub-option includes EPC indication. NSO indication and EPC indication are exclusive.

If the 3GPP-Service-Type Sub-option includes NSO indication, and if 3GPP-APN Sub-option is included, the 3GPP-APN Sub-option is ignored.
5. DHCP Client Behavior

For DHCPv4 protocol, a DHCP client may include a "3GPP-Service" option in the option payload of DHCPDISCOVER and DHCP REQUEST messages being sent toward a DHCP server in order to associate its request with IP services provided by the 3GPP EPC or to explicitly require that its traffic is not sent to the EPC but offloaded.

For DHCPv6 protocol, DHCP client may include a "3GPP-Service" option in the option payload of SOLICIT message being sent toward a DHCP server in order to associate its request with IP services provided by the 3GPP EPC or to explicitly require that its traffic is not sent to the EPC but offloaded.

If the answer from the DHCP server does not include a "3GPP-Service" option, the DHCP client assumes that the answer is not associated with an EPC service.
6. DHCP Server Behavior

A DHCP server that receives a "3GPP-Service" option from a DHCP client but does not support this option, MUST ignore this option and MUST NOT provide this option in the signaling it sends back towards the DHCP client.

Conversely, when a DHCP server has taken this option into account it MUST provide this option in the signaling it sends back towards the DHCP client.

A DHCP server supporting this option SHOULD take the content of this option into account when trying to serve the DHCP client. This may involve making sure relevant signaling (e.g. GTP, PMIP [RFC5213]) is sent to an EPC gateway determined using the parameters of the "3GPP-Service" option.
7. Security Considerations

Security considerations in DHCPv4 are described in [RFC3118].

Security considerations in DHCPv6 are described in [RFC3315].
8. IANA Considerations

IANA is requested to assign one new DHCP option code defined in section 5.
9. Normative References


[TS24008] 3GPP, "Mobile radio interface Layer 3 specification; Core network protocols", 2010.
Authors’ Addresses

Guoyan Liu
ZTE Corporation
No.68 Zijinghua Avenue, Yuhuatai District
Nanjing
China

Phone: +86-25-5287-1362
Email: liu.guoyan@zte.com.cn

Yangwei Tu
ZTE Corporation
No.68 Zijinghua Avenue, Yuhuatai District
Nanjing
China

Phone: +86-25-5287-1362
Email: tu.yangwei@zte.com.cn

Chunhui Zhu
ZTE Corporation
No.68 Zijinghua Avenue, Yuhuatai District
Nanjing
China

Phone: +86-25-5287-4634
Email: zhu.chunhui@zte.com.cn

Wim Hendericks
Alcatel-Lucent

Email: Wim.Henderickx@alcatel-lucent.com

Daniel Derksen
Alcatel-Lucent

Email: Daniel.Derksen@alcatel-lucent.com
Laurent Thiebaut
Alcatel-Lucent

Email: laurent.thiebaut@alcatel-lucent.com
DHCPv6 Failover Design

draft-mrugalski-dhc-dhcpv6-failover-design-01

Abstract

DHCPv6 defined in [RFC3315] does not offer server redundancy. This document defines a design for DHCPv6 failover, a mechanism for running two servers on the same network with capability for either server to take over clients’ leases in case of server failure or network partition. This is a DHCPv6 Failover design document, it is not protocol specification document. It is a second document in a planned series of three documents. DHCPv6 failover requirements are specified in [requirements]. A protocol specification document is planned to follow 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 September 13, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents.
carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Requirements Language .......................................... 4
2. Glossary ................................................................. 4
3. Introduction ............................................................. 5
   3.1. Additional Requirements ........................................... 5
   3.2. Features out of Scope: Load Balancing .......................... 6
4. Protocol Overview ..................................................... 6
   4.1. Failover Machine State Overview ................................. 7
5. Connection Management ................................................ 9
   5.1. Creating Connections ............................................... 9
   5.2. Endpoint Identification .......................................... 10
6. Resource Allocation ................................................ 11
   6.1. Proportional Allocation ........................................... 12
   6.2. Independent Allocation ........................................... 13
   6.3. Determining Allocation Approach .................................. 13
      6.3.1. IPv6 Addresses ............................................... 13
      6.3.2. IPv6 Prefixes ............................................... 13
7. Failover Mechanisms .................................................. 13
   7.1. Time Skew ............................................................ 14
   7.2. Time expression .................................................... 14
   7.3. Lazy updates ........................................................ 14
   7.4. MCLT concept ........................................................ 15
      7.4.1. MCLT example .................................................. 16
   7.5. Unreachability detection ......................................... 17
   7.6. Re-allocating Leases .............................................. 17
   7.7. Sending Data ........................................................ 18
      7.7.1. Required Data .................................................. 18
      7.7.2. Optional Data ................................................ 18
   7.8. Receiving Data .................................................... 18
      7.8.1. Conflict Resolution ......................................... 18
      7.8.2. Acknowledging Reception ..................................... 19
8. Endpoint States ........................................................ 19
   8.1. State Machine Operation .......................................... 19
   8.2. State Machine Initialization ..................................... 22
   8.3. STARTUP State ...................................................... 22
      8.3.1. Operation in STARTUP State ................................... 22
      8.3.2. Transition Out of STARTUP State ............................... 22
   8.4. PARTNER-DOWN State .............................................. 24
      8.4.1. Operation in PARTNER-DOWN State ............................... 24
      8.4.2. Transition Out of PARTNER-DOWN State ........................ 24
8.5. RECOVER State ........................................ 25
  8.5.1. Operation in RECOVER State ....................... 25
  8.5.2. Transition Out of RECOVER State .................. 25
8.6. RECOVER-WAIT State .................................... 27
  8.6.1. Operation in RECOVER-WAIT State ................ 28
  8.6.2. Transition Out of RECOVER-WAIT State ............ 28
8.7. RECOVER-DONE State .................................... 28
  8.7.1. Operation in RECOVER-DONE State ................. 29
  8.7.2. Transition Out of RECOVER-DONE State ............ 29
8.8. NORMAL State .......................................... 29
  8.8.1. Operation in NORMAL State ....................... 29
  8.8.2. Transition Out of NORMAL State ................. 30
8.9. COMMUNICATIONS-INTERRUPTED State .................... 31
  8.9.1. Operation in COMMUNICATIONS-INTERRUPTED State .. 31
  8.9.2. Transition Out of COMMUNICATIONS-INTERRUPTED State 32
8.10. POTENTIAL-CONFLICT State ............................. 33
  8.10.1. Operation in POTENTIAL-CONFLICT State ......... 34
  8.10.2. Transition Out of POTENTIAL-CONFLICT State .... 34
8.11. RESOLUTION-INTERRUPTED State .......................... 35
  8.11.1. Operation in RESOLUTION-INTERRUPTED State .... 36
  8.11.2. Transition Out of RESOLUTION-INTERRUPTED State .. 36
8.12. CONFLICT-DONE State .................................. 36
  8.12.1. Operation in CONFLICT-DONE State .............. 37
  8.12.2. Transition Out of CONFLICT-DONE State ......... 37
8.13. PAUSED State ......................................... 37
  8.13.1. Operation in PAUSED State ..................... 37
  8.13.2. Transition Out of PAUSED State ................ 38
8.14. SHUTDOWN State ....................................... 38
  8.14.2. Transition Out of SHUTDOWN State ............ 38
9. Proposed extensions ....................................... 38
  9.1. Active-active mode .................................. 39
10. Dynamic DNS Considerations .............................. 39
11. Reservations and failover ............................... 39
12. Protocol entities ....................................... 39
  12.1. Failover Protocol .................................. 40
  12.2. Protocol constants .................................. 40
13. Open questions ......................................... 40
14. Security Considerations .................................. 40
15. IANA Considerations .................................... 40
16. Acknowledgements ....................................... 41
17. References ............................................ 41
  17.1. Normative References ................................ 41
  17.2. Informative References .............................. 41
Authors’ Addresses ......................................... 42
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. Glossary

This is a supplemental glossary that should be combined with definitions in Section 3 of [requirements].

- **Failover endpoint** - The failover protocol allows for there to be a unique failover 'endpoint' per partner per role per relationship (where role is primary or secondary and the relationship is defined by the relationship-name). This failover endpoint can take actions and hold unique states. Typically, there is a one failover endpoint per partner (server), although there may be more. 'Server' and 'failover endpoint' are synonymous only if the server participates in only one failover relationship. However, for the sake of simplicity 'Server' is used throughout the document to refer to a failover endpoint unless to do so would be confusing.

- **Failover transmission** - all messages exchanged between partners.

- **Independent Allocation** - a prefix allocation algorithm to split the available pool of resources between the primary and secondary servers that is particularly well suited for vast pools (i.e. when available resources are not expected to deplete). See Section 6.2 for details.

- **Primary Server**

- **Proportional Allocation** - a prefix allocation algorithm to split the available free leases between the primary and secondary servers that is particularly well suited for more limited resources. See Section 6.1 for details.

- **Resource** - an IPv6 address or a IPv6 prefix.

- **Responsive** - A server that is responsive, will respond to DHCPv6 client requests.

- **Secondary Server**

- **Server** - A DHCPv6 server that implements DHCPv6 failover. 'Server' and 'failover endpoint' as synonymous only if server
participates in only one failover relationship.

- Unresponsive - A server that is unresponsive will not respond to DHCPv6 client requests.

3. Introduction

The failover protocol design provides a means for cooperating DHCPv6 servers to work together to provide a DHCPv6 service with availability that is increased beyond that which could be provided by a single DHCPv6 server operating alone. It is designed to protect DHCPv6 clients against server unreachability, including server failure and network partition. It is possible to deploy exactly two servers that are able to continue providing a lease on an IPv6 address or on an IPv6 prefix without the DHCPv6 client experiencing lease expiration or a reassignment of a lease to a different IPv6 address in the event of failure by one or the other of the two servers.

This protocol defines active-passive mode, sometimes also called hot standby model. This means that during normal operation one server is active (i.e. actively responds to clients’ requests) while the second is passive (i.e. it does receive clients’ requests, but does not respond to them and only maintains a copy of lease database and is ready to take over incoming queries in case of primary server failure). Active-active mode (i.e. both servers actively handling clients’ requests) is currently not supported for the sake of simplicity. Such mode may be defined as an extension at a later time.

The failover protocol is designed to provide lease stability for leases with lease times beyond a short period. Due to the additional overhead required, failover is not suitable for leases shorter than 30 seconds. The DHCPv6 Failover protocol MUST NOT be used for leases shorter than 30 seconds.

This design attempts to fulfill all DHCPv6 failover requirements defined in [requirements].

3.1. Additional Requirements

The following requirements are not related to failover mechanism in general, but rather to this particular design.

1. Minimize Asymmetry - while there are two distinct roles in failover (primary and secondary server), the differences between those two roles should be as small as possible. This will yield a simpler design as well as a simpler implementation of that
3.2. Features out of Scope: Load Balancing

It may be tempting to extend DHCPv6 failover mechanism to also offer load balancing, as DHCPv4 failover did. Here is the reasoning for this decision. In general case (not related to failover) load balancing solutions are used when each server is not able to handle total incoming traffic. However, by the very definition, DHCPv6 failover is supposed to assume service availability despite failure of one server. That leads to conclusion that each server must be able to handle whole traffic. Therefore in properly provisioned setup, load balancing is not needed.

4. Protocol Overview

The DHCPv6 Failover Protocol is defined as a communication between failover partners with all associated algorithms and mechanisms. Failover communication is conducted over a TCP connection established between the partners. The protocol reuses the framing format specified in Section 5.1 of DHCPv6 Bulk Leasequery [RFC5460], but uses different message types. Additional failover-specific message types will be defined. All information is sent over the connection as typical DHCPv6 Options, following format defined in Section 22.1 of [RFC3315].

After initialization, the primary server establishes a TCP connection with its partner. The primary server sends a CONNECT message with initial parameters. Secondary server responds with CONNECTACK.

Depending on the failover state of each partner, they MUST initiate one of the binding update procedures. Each server MAY send an UPDREQ message to request its partner to send all updates that have not been sent yet (this case applies when partner has an existing database and wants to update it). Alternatively, a server MAY choose to send an UPDREQALL message to request a full lease database transmission including all leases (this case applies in case of booting up new server after installation, corruption or complete loss of database, or other catastrophic failure).

Servers exchange lease information by using BNDUPD messages. Depending on local and remote state of a lease, a server may either accept or reject the update. Reception of lease update information is confirmed by responding with BNDACK message with appropriate status. The majority of the messages sent over a failover TCP connection consists of BNDUPD and BNDACK messages.
A subset of available resources (addresses or prefixes) is reserved for secondary server use. This is required for handling a case where both servers are able to communicate with clients, but unable to communicate with each other. After initial connection is established, the secondary server requests a pool of available addresses by sending a POOLREQ message. The primary server assigns a pool to the secondary by transmitting a POOLRESP message and then sending a series of BNDUPD messages. The secondary server may initiate such pool request at any time when maintaining communication with primary server.

Failover servers use a lazy update mechanism to update their failover partner about changes to their lease state database. After a server performs any modifications to its lease state database (assign a new lease, extend an existing one, release or expire a lease), it sends its response to the client’s request first (performing the "regular" DHCPv6 operation) and then informs its failover partner using a BNDUPD message. This BNDUPD message SHOULD be sent soon after the response is sent to the DHCPv6 client, but there is no specific requirement of a minimum time in which to do so.

The major problem with lazy update mechanism is the case when the server crashes after sending response to client, but before sending the lazy update to its partner (or when communication between partners is interrupted). To solve this problem, concept known as the Maximum Client Lead Time (MCLT) (initially designed for DHCPv4 failover) is used. The MCLT is the maximum amount of time that one server can extend a lease for a client’s binding beyond the time known by its failover partner. See Section 7.4 for detailed description how MCLT affects assigned lease times.

Servers verify each others availability by periodically exchanging CONTACT messages. See Section 7.5 for discussion about detecting partner’s unreachability.

A server that is being shut down transmits a DISCONNECT message, closes the connection with its failover partner and stops operation. A Server SHOULD transmit any pending lease updates before transmitting DISCONNECT message.

4.1. Failover Machine State Overview

The following section provides simplified description of all states. For the sake of clarity and simplicity, it omits important details. For complete description, see Section 8. In case of a disagreement between simplified and complete description, please follow Section 8.

Each server may be in one of the well defines states. In each state
a server may be either responsive (responds to clients’ queries) or unresponsive (clients’ queries are ignored).

A server starts its operation in short-lived STARTUP state. A server determines its partner reachability and state and usually returns back to the state it was in before shutdown.

During typical operation when servers maintain communication, both are in NORMAL state. In that state only primary responds to clients’ requests. A secondary server in unresponsive.

If a server discovers that its partner is no longer reachable, it goes to COMMUNICATIONS-INTERRUPTED state. Server must be extra cautious as it can’t distinguish if its partner is down or just communication between servers is interrupted. Since communication between partners is not possible, a server must act on the assumption that if its partner is up, it follows defined procedure. In particular, not extend any lease beyond its partner knowledge by at most MCLT. That imposes additional burden on the server. Therefore it is not recommended to operate for prolonged periods in this state. Once communication is reestablished, server may go into NORMAL, POTENTIAL-CONFLICT or PARTNER-DOWN state. It may also stay in COMMUNICATIONS-INTERRUPTED if certain conditions are met.

Once a server is switched into PARTNER-DOWN (when auto-partner-down is used or as a result of administrative action), it can extend leases, regardless of the original server that initially granted the lease. In that state server handles leases from its own pool, but is also able to serve pool from its downed partner. MCLT restrictions no longer apply. Operation in this mode is less demanding for the server that remains operational, than in COMMUNICATIONS-INTERRUPTED state, but PARTNER-DOWN does not offer any kind of redundancy.

When server loses its database (e.g. due to first time run or catastrophic failure) or detects that is partner is in PARTNER-DOWN state and additional conditions are met, it switches to RECOVER state. In that state server acknowledges that content of its database is doubtful and needs to refresh its database from its partner. Once this operation is done, it switches to RECOVER-WAIT and later to RECOVER-DONE.

Once servers reestablish connection, they discover each others’ state. Depending on the conditions, they may return to NORMAL or move to POTENTIAL-CONFLICT in case of unexpected partner’s state. It is a goal of this protocol to minimize the possibility that POTENTIAL-CONFLICT state is ever entered. Servers running in POTENTIAL-CONFLICT do not respond to clients’ requests and work on resolving potential conflicts. Once outstanding lease updates are
exchanged, servers move to CONFLICT-DONE or NORMAL states.

Servers that are recovering from potential conflict and loose communication, switch to RESOLUTION-INTERRUPTED.

Server that is being shut down, switches briefly to SHUTDOWN state and communicates its state to its partner before actual termination.

5. Connection Management

5.1. Creating Connections

Every server implementing the failover protocol SHOULD attempt to connect to all of its partners periodically, where the period is implementation dependent and SHOULD be configurable. In the event that a connection has been rejected by a CONNECTACK message with a reject-reason option contained in it or a DISCONNECT message, a server SHOULD reduce the frequency with which it attempts to connect to that server but it SHOULD continue to attempt to connect periodically.

When a connection attempt succeeds, if the server generating the connection attempt is a primary server for that relationship, then it MUST send a CONNECT message down the connection. If it is not a primary server for the relationship, then it MUST just drop the connection and wait for the primary server to connect to it.

When a connection attempt is received, the only information that the receiving server has is the IP address of the partner initiating a connection. It also knows whether it has the primary role for any failover relationships with the connecting server. If it has any relationships for which it is a primary server, it should initiate a connection of its own to the partner server, one for each primary relationship it has with that server.

If it has any relationships with the connecting server for which it is a secondary server, it should just await the CONNECT message to determine which relationship this connection is to serve.

If it has no secondary relationships with the connecting server, it SHOULD drop the connection.

To summarize -- a primary server MUST use a connection that it has initiated in order to send a CONNECT message. Every server that is a secondary server in a relationship attempts to create a connection to the server which is primary in the relationship, but that connection is only used to stimulate the primary server into recognizing that
the secondary server is ready for operation. The reason behind this is that the secondary server has no way to communicate to the primary server which relationship a connection is designed to serve.

A server which has multiple secondary relationships with a primary server SHOULD only send one stimulus connection attempt to the primary server.

Once a connection is established, the primary server MUST send a CONNECT message across the connection. A secondary server MUST wait for the CONNECT message from a primary server. If the secondary server doesn’t receive a CONNECT message from the primary server in an installation dependent amount of time, it MAY drop the connection and send another stimulus connection attempt to the primary server.

Every CONNECT message includes a TLS-request option, and if the CONNECTACK message does not reject the CONNECT message and the TLS-reply option says TLS MUST be used, then the servers will immediately enter into TLS negotiation.

Once TLS negotiation is complete, the primary server MUST resend the CONNECT message on the newly secured TLS connection and then wait for the CONNECTACK message in response. The TLS-request and TLS-reply options MUST NOT appear in either this second CONNECT or its associated CONNECTACK message as they had in the first messages.

The second message sent over a new connection (either a bare TCP connection or a connection utilizing TLS) is a STATE message. Upon the receipt of this message, the receiver can consider communications up.

A secondary server MUST NOT respond to the closing of a TCP connection with a blind attempt to reconnect -- there may be another TCP connection to the same failover partner already in use.

5.2. Endpoint Identification

The proper operation of the failover protocol requires more than the transmission of messages between one server and the other. Each endpoint might seem to be a single DHCPv6 server, but in fact there are situations where additional flexibility in configuration is useful. A failover endpoint is always associated with a set of DHCPv6 prefixes that are configured on the DHCPv6 server where the endpoint appears. A DHCPv6 prefix MUST NOT be associated with more than one failover endpoint.

The failover protocol SHOULD be configured with one failover relationship between each pair of failover servers. In this case
there is one failover endpoint for that relationship on each failover partner. This failover relationship MUST have a unique name.

There is typically little need for additional relationships between any two servers but there MAY be more than one failover relationship between two servers -- however each MUST have a unique relationship name.

Any failover endpoint can take actions and hold unique states.

This document frequently describes the behavior of the protocol in terms of primary and secondary servers, not primary and secondary failover endpoints. However, it is important to remember that every 'server' described in this document is in reality a failover endpoint that resides in a particular process, and that several failover endpoints may reside in the same server process.

It is not the case that there is a unique failover endpoint for each prefix that participates in a failover relationship. On one server, there is (typically) one failover endpoint per partner, regardless of how many prefixes are managed by that combination of partner and role. Conversely, on a particular server, any given prefix will be associated with exactly one failover endpoint.

When a connection is received from the partner, the unique failover endpoint to which the message is directed is determined solely by the IP address of the partner, the relationship-name, and the role of the receiving server.

6. Resource Allocation

Currently there are two allocation algorithms defined for resources (addresses or prefixes). Additional allocation schemes may be defined as future extensions.

1. Proportional Allocation - This allocation algorithm is a direct application of algorithm defined in [dhcpv4-failover] to DHCPv6. Available resources are split between primary and secondary server. Released resources are always returned to primary server. Primary and secondary servers may initiate a rebalancing procedure, when disparity between resources available to each server reaches a preconfigured threshold. Only resources that are not leased to any clients are "owned" by one of the servers. This algorithm is particularly well suited for scenarios where amount of available resources is limited, as may be the case for prefix delegation. See Section 6.1 for details.
2. Independent Allocation - This allocation algorithm assumes that available resources are split between primary and secondary servers as well. In this case, however, resources are assigned to a specific server for all time, regardless if they are available or currently used. This algorithm is much simpler than proportional allocation, because resource imbalance doesn't have to be checked and there is no rebalancing for independent allocation. This algorithm is particularly well suited for scenarios where there is an abundance of available resources which is typically the case for DHCPv6 address allocation. See Section 6.2 for details.

6.1. Proportional Allocation

In this allocation scheme, each server has its own pool of available resources. Note that a resource is not "owned" by a particular server throughout its entire lifetime. Only a resource which is available is "owned" by a particular server -- once it has been leased to a client, it is not owned by either failover partner. When it finally becomes available again, it will be owned initially by the primary server, and it may or may not be allocated to the secondary server by the primary server.

So, the flow of a resource is as follows: initially a resource is owned by the primary server. It may be allocated to the secondary server if it is available, and then it is owned by the secondary server. Either server can allocate available resources which they own to clients, in which case they cease to own them. When the client releases the resource or the lease on it expires, it will again become available and will be owned by the primary.

A resource will not become owned by the server which allocated it initially when it is released or the lease expires because, in general, that server will have had to replenish its pool of available resources well in advance of any likely lease expirations. Thus, having a particular resource cycle back to the secondary might well put the secondary more out of balance with respect to the primary instead of enhancing the balance of available addresses or prefixes between them.

TODO: Need to rework this v4-specific vocabulary to v6, once we decide how things will look like in v6.

When they are used, these proportional pools are used for allocation when in every state but PARTNER-DOWN state. In PARTNER-DOWN state a failover server can allocate from either pool. This allocation and maintenance of these address pools is an area of some sensitivity, since the goal is to maintain a more or less constant ratio of
available addresses between the two servers.

TODO: Reuse rest of the description from section 5.4 from [dhcpv4-failover] here.

6.2. Independent Allocation

In this allocation scheme, available resources are split between servers. Available resources are split between the primary and secondary servers as part of initial connection establishment. Once resources are allocated to each server, there is no need to reassign them. This algorithm is simpler than proportional allocation since it requires no less initial communication and does not require a rebalancing mechanism, but it assumes that the pool assigned to each server will never deplete. That is often a reasonable assumption for IPv6 addresses (e.g. servers are often assigned a /64 pool that contains many more addresses than existing electronic devices on Earth). This allocation mechanism SHOULD be used for IPv6 addresses, unless configured address pool is small or is otherwise administratively limited.

Once each server is assigned a resource pool during initial connection establishment, it may allocate assigned resources to clients. Once a client release a resource or its lease is expired, the returned resource returns to pool for the same server. Resources never changes servers.

During COMMUNICATION-INTERRUPTED events, a partner MAY continue extending existing leases when requested by clients. A healthy partner MUST NOT lease resources that were assigned to its downed partner and later released by a client unless it is in PARTNER-DOWN state.

6.3. Determining Allocation Approach

6.3.1. IPv6 Addresses

6.3.2. IPv6 Prefixes

7. Failover Mechanisms

This section lays out an overview of the communication between partners and other mechanisms required for failover operation. As this is a design document, not a protocol specification, high level ideas are presented without implementation specific details (e.g. lack of on-wire formats). Implementation details will be specified in a separate draft.
7.1. Time Skew

Partners exchange information about known lease states. To reliably compare a known lease state with an update received from a partner, servers must be able to reliably compare the times stored in the known lease state with the times received in the update. Although a simple approach would be to require both partners to use synchronized time, e.g. by using NTP, such a service may become unavailable in some scenarios that failover expects to cover, e.g. network partition. Therefore a mechanism to measure and track relative time differences between servers is necessary. To do so, each message MUST contain FO_TIMESTAMP option that contains the timestamp of the transmission in the time context of the transmitter. The transmitting server MUST set this as close to the actual transmission as possible. The receiving partner MUST store its own timestamp of reception event as close to the actual reception as possible. The received timestamp information is then compared with local timestamp.

To account for packet delay variation (jitter), the measured difference is not used directly, but rather the moving average of last TIME_SKEW_PKTS_AVG packets time difference is calculated. This averaged value is referred to as the time skew. Note that the time skew algorithm allows cooperation between clients with completely desynchronized clocks as well as those whose desynchronization itself is not constant.

7.2. Time expression

Timestamps are expressed as number of seconds since midnight (UTC), January 1, 2000, modulo 2^32. Note: that is the same approach as used in creation of DUID-LLT (see Section 9.2 of [RFC3315]).

Time differences are expressed in seconds and are signed.

7.3. Lazy updates

Lazy update refers to the requirement placed on a server implementing a failover protocol to update its failover partner whenever the binding database changes. A failover protocol which didn’t support lazy update would require the failover partner update to complete before a DHCPv6 server could respond to a DHCPv6 client request. The lazy update mechanism allows a server to allocate a new or extend an existing lease and then update its failover partner as time permits.

Although the lazy update mechanism does not introduce additional delays in server response times, it introduces other difficulties. The key problem with lazy update is that when a server fails after updating a client with a particular lease time and before updating
its partner, the partner will believe that a lease has expired even though the client still retains a valid lease on that address or prefix.

7.4. MCLT concept

In order to handle problem introduced by lazy updates (see Section 7.3), a period of time known as the "Maximum Client Lead Time" (MCLT) is defined and must be known to both the primary and secondary servers. Proper use of this time interval places an upper bound on the difference allowed between the lease time provided to a DHCPv6 client by a server and the lease time known by that server’s failover partner.

The MCLT is typically much less than the lease time that a server has been configured to offer a client, and so some strategy must exist to allow a server to offer the configured lease time to a client. During a lazy update the updating server typically updates its partner with a potential expiration time which is longer than the lease time previously given to the client and which is longer than the lease time that the server has been configured to give a client. This allows that server to give a longer lease time to the client the next time the client renews its lease, since the time that it will give to the client will not exceed the MCLT beyond the potential expiration time acknowledged by its partner.

The fundamental relationship on which much of the correctness of this protocol depends is that the lease expiration time known to a DHCPv6 client MUST NOT under any circumstances be more than the maximum client lead time (MCLT) greater than the potential expiration time known to a server’s partner.

The remainder of this section makes the above fundamental relationship more explicit.

This protocol requires a DHCPv6 server to deal with several different lease intervals and places specific restrictions on their relationships. The purpose of these restrictions is to allow the other server in the pair to be able to make certain assumptions in the absence of an ability to communicate between servers.

The different times are:

**desired valid lifetime:**

The desired valid lifetime is the lease interval that a DHCPv6 server would like to give to a DHCPv6 client in the absence of any restrictions imposed by the failover protocol. Its determination is outside of the scope of this protocol. Typically this is the
result of external configuration of a DHCPv6 server.

actual valid lifetime:
The actual valid lifetime is the lease interval that a DHCPv6 server gives out to a DHCPv6 client. It may be shorter than the desired valid lifetime (as explained below).

potential valid lifetime:
The potential valid lifetime is the potential lease expiration interval the local server tells to its partner in a BNDUPD message.

acknowledged potential valid lifetime:
The acknowledged potential valid lifetime is the potential lease interval the partner server has most recently acknowledged in a BNDACK message.

7.4.1. MCLT example

The following example demonstrates the MCLT concept in practice. The values used are arbitrarily chosen are and not a recommendation for actual values. The MCLT in this case is 1 hour. The desired valid lifetime is 3 days, and its renewal time is half the valid lifetime.

When a server makes an offer for a new lease on an IP address to a DHCPv6 client, it determines the desired valid lifetime (in this case, 3 days). It then examines the acknowledged potential valid lifetime (which in this case is zero) and determines the remainder of the time left to run, which is also zero. To this it adds the MCLT. Since the actual valid lifetime cannot be allowed to exceed the remainder of the current acknowledged potential valid lifetime plus the MCLT, the offer made to the client is for the remainder of the current acknowledged potential valid lifetime (i.e., zero) plus the MCLT. Thus, the actual valid lifetime is 1 hour.

Once the server has sent the REPLY to the DHCPv6 client, it will update its failover partner with the lease information. However, the desired potential valid lifetime will be composed of one half of the current actual valid lifetime added to the desired valid lifetime. Thus, the failover partner is updated with a BNDUPD with a potential valid lifetime of 3 days + 1/2 hour.

When the primary server receives a BNDACK to its update of the secondary server's (partner's) potential valid lifetime, it records that as the acknowledged potential valid lifetime. A server MUST NOT send a BNDACK in response to a BNDUPD message until it is sure that the information in the BNDUPD message has been updated in its lease database. Thus, the primary server in this case can be sure that the
secondary server has recorded the potential lease interval in its stable storage when the primary server receives a BNDACK message from the secondary server.

When the DHCPv6 client attempts to renew at T1 (approximately one half an hour from the start of the lease), the primary server again determines the desired valid lifetime, which is still 3 days. It then compares this with the remaining acknowledged potential valid lifetime (3 days + 1/2 hour) and adjusts for the time passed since the secondary was last updated (1/2 hour). Thus the time remaining of the acknowledged potential valid interval is 3 days. Adding the MCLT to this yields 3 days plus 1 hour, which is more than the desired valid lifetime of 3 days. So the client is renewed for the desired valid lifetime -- 3 days.

When the primary DHCPv6 server updates the secondary DHCPv6 server after the DHCPv6 client’s renewal REPLY is complete, it will calculate the desired potential valid lifetime as the T1 fraction of the actual client valid lifetime (1/2 of 3 days this time = 1.5 days). To this it will add the desired client valid lifetime of 3 days, yielding a total desired potential valid lifetime of 4.5 days. In this way, the primary attempts to have the secondary always "lead" the client in its understanding of the client’s valid lifetime so as to be able to always offer the client the desired client valid lifetime.

Once the initial actual client valid lifetime of the MCLT is past, the protocol operates effectively like the DHCPv6 protocol does today in its behavior concerning valid lifetimes. However, the guarantee that the actual client valid lifetime will never exceed the remaining acknowledged partner server potential valid lifetime by more than the MCLT allows full recovery from a variety of failures.

7.5. Unreachability detection

Each partner maintains an FO_SEND timer for each partner connection. The FO_SEND timer is reset every time any message is transmitted. If the timer reaches the FO_SEND_MAX value, a CONTACT message is transmitted and timer is reset. The CONTACT message may be transmitted at any time.

Discussion: Perhaps it would be more reasonable to use echo-reply approach, rather than periodic transmissions?

7.6. Re-allocating Leases

TODO: Describe controlled re-allocation of released/expired leases to different clients.
7.7. Sending Data

Each server updates its failover partner about recent changes in lease states. Each update must include following information:

1. resource type - non-temporary address or a prefix
2. resource information - actual address or prefix
3. valid life time requested by client
4. IAID - Identity Association used by client, while obtaining this lease. (Note1: one client may use many IAID simultaneously. Note2: IAID for IA, TA and PD are orthogonal number spaces.)
5. valid life time sent to client
6. potential valid life time
7. preferred life time sent to client
8. CLTT - Client Last Transaction Time, a timestamp of the last received transmission from a client
9. assigned FQDN names, if any (optional)

Discussion: Do we need T1 as well? Something like next expected client transmission?

Q: Maybe we could reuse IA_NA and IA_PD options here? Yes.

Q: Do we care about preferred lifetime? (presumably no). Certainly not what was requested by the client.

Q: Do we care about IAID? (presumably yes) Yes.

7.7.1. Required Data

7.7.2. Optional Data

7.8. Receiving Data

7.8.1. Conflict Resolution

TODO: This is just a loose collection of notes. This section will probably need to be rewritten as a a flowchart of some kind.

The server receiving a lease update from its partner must evaluate
the received lease information to see if it is consistent with already known state and decide which information - previously known or just received - is "better". The server should take into consideration the following aspects: if the lease is already assigned to specific client, who had contact with client recently, start time of the lease, etc.

The lease update may be accepted or rejected. Rejection SHOULD NOT change the flag in a lease that says that it should be transmitted to the failover partner. If this flag is set, then it should be transmitted, but if it is not already set, the rejection of a lease state update SHOULD NOT trigger an automatic update of the failover partner sending the rejected update. The potential for update storms is too great, and in the unusual case where the servers simply can’t agree, that disagreement is better than an update storm.

Discussion: There will definitely be different types of update rejections. For example, this will allow a server to treat differently a case when receiving a new lease that it previously haven’t seen than a case when partner sends old version of a lease for which a newer state is known.

7.8.2. Acknowledging Reception

8. Endpoint States

8.1. State Machine Operation

Each server (or, more accurately, failover endpoint) can take on a variety of failover states. These states play a crucial role in determining the actions that a server will perform when processing a request from a DHCPv6 client as well as dealing with changing external conditions (e.g., loss of connection to a failover partner).

The failover state in which a server is running controls the following behaviors:

- Responsiveness -- the server is either responsive to DHCPv6 client requests or it is not.
- Allocation Pool -- which pool of addresses (or prefixes) can be used for allocation on receipt of a SOLICIT message.
- MCLT -- ensure that valid lifetimes are not beyond what the partner has acked plus the MCLT (or not).

A server will transition from one failover state to another based on
the specific values held by the following state variables:

- Current failover state.
- Communications status (OK or not OK).
- Partner’s failover state (if known).

Whenever the either of the last two of the above state variables changes state, the state machine is invoked, which may then trigger a change in the current failover state. Thus, whenever the communications status changes, the state machine is processing is invoked. This may or may not result in a change in the current failover state.

Whenever a server transitions to a new failover state, the new state MUST be communicated to its failover partner in a STATE message if the communications status is OK. In addition, whenever a server makes a transition into a new state, it MUST record the new state, its current understanding of its partner’s state, and the time at which it entered the new state in stable storage.

The following state transition diagram gives a condensed view of the state machine. If there is a difference between the words describing a particular state and the diagram below, the words should be considered authoritative.

A transition into SHUTDOWN or PAUSED state is not represented in the following figure, since other than sending that state to its partner, the remaining actions involved look just like the server halting in its otherwise current state, which then becomes the previous state upon server restart.
Figure 1: Failover Endpoint State Machine
8.2. State Machine Initialization

TODO

8.3. STARTUP State

The STARTUP state affords an opportunity for a server to probe its partner server, before starting to service DHCP clients. When in the STARTUP state, a server attempts to learn its partner’s state and determine (using that information if it is available) what state it should enter.

The STARTUP state is not shown with any specific state transitions in the state machine diagram (Figure 1) because the processing during the STARTUP state can cause the server to transition to any of the other states, so that specific state transition arcs would only obscure other information.

8.3.1. Operation in STARTUP State

The server MUST NOT be responsive in STARTUP state.

Whenever a STATE message is sent to the partner while in STARTUP state the STARTUP flag MUST be set the message and the previously recorded failover state MUST be placed in the server-state option.

8.3.2. Transition Out of STARTUP State

The following algorithm is followed every time the server initializes itself, and enters STARTUP state.

Step 1:

If there is any record in stable storage of a previous failover state for this server, set PREVIOUS-STATE to the last recorded value in stable storage, and go to Step 2.

If there is no record of any previous failover state in stable storage for this server, then set the PREVIOUS-STATE to RECOVER and set the TIME-OF-FAILURE to 0. This will allow two servers which already have lease information to synchronize themselves prior to operating.

In some cases, an existing server will be commissioned as a failover server and brought back into operation where its partner is not yet available. In this case, the newly commissioned failover server will not operate until its partner comes online -- but it has operational responsibilities as a DHCP server nonetheless. To properly handle
this situation, a server SHOULD be configurable in such a way as to move directly into PARTNER-DOWN state after the startup period expires if it has been unable to contact its partner during the startup period.

Step 2:

If the previous state is one where communications was "OK", then set the previous state to the state that is the result of the communications failed state transition (if such transition exists -- some states don’t have a communications failed state transition, since they allow both communications OK and failed).

Step 3:

Start the STARTUP state timer. The time that a server remains in the STARTUP state (absent any communications with its partner) is implementation dependent but SHOULD be short. It SHOULD be long enough for a TCP connection to be created to a heavily loaded partner across a slow network.

Step 4:

Attempt to create a TCP connection to the failover partner.

Step 5:

Wait for "communications OK".

When and if communications become "okay", clear the STARTUP flag, and set the current state to the PREVIOUS-STATE.

If the partner is in PARTNER-DOWN state, and if the time at which it entered PARTNER-DOWN state (as received in the start-time-of-state option in the STATE message) is later than the last recorded time of operation of this server, then set CURRENT-STATE to RECOVER. If the time at which it entered PARTNER-DOWN state is earlier than the last recorded time of operation of this server, then set CURRENT-STATE to POTENTIAL-CONFLICT.

Then, transition to the current state and take the "communications OK" state transition based on the current state of this server and the partner.

Step 6:

If the startup time expires the server SHOULD go transition to the PREVIOUS-STATE.
8.4. PARTNER-DOWN State

PARTNER-DOWN state is a state either server can enter. When in this state, the server assumes that it is the only server operating and serving the client base. If one server is in PARTNER-DOWN state, the other server MUST NOT be operating.

8.4.1. Operation in PARTNER-DOWN State

The server MUST be responsive in PARTNER-DOWN state.

It will allow renewal of all outstanding leases on IP addresses. For those IP addresses for which the server is using proportional allocation, it will allocate IP addresses from its own pool, and after a fixed period of time (the MCLT interval) has elapsed from entry into PARTNER-DOWN state, it will allocate IP addresses from the set of all available IP addresses.

Any IP address tagged as available for allocation by the other server (at entry to PARTNER-DOWN state) MUST NOT be allocated to a new client until the maximum-client-lead-time beyond the entry into PARTNER-DOWN state has elapsed.

A server in PARTNER-DOWN state MUST NOT allocate an IP address to a DHCP client different from that to which it was allocated at the entrance to PARTNER-DOWN state until the maximum-client-lead-time beyond the maximum of the following times: client expiration time, most recently transmitted potential-expiration-time, most recently received ack of potential-expiration-time from the partner, and most recently acked potential-expiration-time to the partner. See section 7.1.5 for details. If this time would be earlier than the current time plus the maximum-client-lead-time, then the time the server entered PARTNER-DOWN state plus the maximum-client-lead-time is used.

The server is not restricted by the MCLT when offering lease times while in PARTNER-DOWN state.

8.4.2. Transition Out of PARTNER-DOWN State

When a server in PARTNER-DOWN state succeeds in establishing a connection to its partner, its actions are conditional on the state and flags received in the STATE message from the other server as part of the process of establishing the connection.

If the STARTUP bit is set in the server-flags option of a received STATE message, a server in PARTNER-DOWN state MUST NOT take any state transitions based on reestablishing communications. Essentially, if a server is in PARTNER-DOWN state, it ignores all STATE messages from
its partner that have the STARTUP bit set in the server-flags option of the STATE message. THIS NEEDS TO BE MOVED

If the STARTUP bit is not set in the server-flags option of a STATE message received from its partner, then a server in PARTNER-DOWN state takes the following actions based on the state of the partner as received in a STATE message (either immediately after establishing communications or at any time later when a new state is received)

If the partner is in:

NORMAL, COMMUNICATIONS-INTERRUPTED, PARTNER-DOWN, POTENTIAL-CONFLICT, RESOLUTION-INTERRUPTED, or CONFLICT-DONE state

transition to POTENTIAL-CONFLICT state

If the partner is in:

RECOVER, RECOVER-WAIT, SHUTDOWN, PAUSED state

stay in PARTNER-DOWN state

If the partner is in:

RECOVER-DONE state

transition into NORMAL state

8.5. RECOVER State

This state indicates that the server has no information in its stable storage or that it is re-integrating with a server in PARTNER-DOWN state after it has been down. A server in this state MUST attempt to refresh its stable storage from the other server.

8.5.1. Operation in RECOVER State

The server MUST NOT be responsive in RECOVER state.

A server in RECOVER state will attempt to reestablish communications with the other server.

8.5.2. Transition Out of RECOVER State

If the other server is in POTENTIAL-CONFLICT, RESOLUTION-INTERRUPTED, or CONFLICT-DONE state when communications are reestablished, then the server in RECOVER state will move to POTENTIAL-CONFLICT state itself.
If the other server is in any other state, then the server in RECOVER state will request an update of missing binding information by sending an UPDREQ message. If the server has determined that it has lost its stable storage because it has no record of ever having talked to its partner, while its partner does have a record of communicating with it, it MUST send an UPDREQALL message, otherwise it MUST send an UPDREQ message.

It will wait for an UPDDONE message, and upon receipt of that message it will transition to RECOVER-WAIT state.

If communications fails during the reception of the results of the UPDREQ or UPDREQALL message, the server will remain in RECOVER state, and will re-issue the UPDREQ or UPDREQALL when communications are re-established.

If an UPDDONE message isn’t received within an implementation dependent amount of time, and no BNDUPD messages are being received, the connection SHOULD be dropped.
If, at any time while a server is in RECOVER state communications fails, the server will stay in RECOVER state. When communications are restored, it will restart the process of transitioning out of RECOVER state.

### 8.6. RECOVER-WAIT State

This state indicates that the server has done an UPDREQ or UPDREQALL and has received the UPDDONE message indicating that it has received all outstanding binding update information. In the RECOVER-WAIT state the server will wait for the MCLT in order to ensure that any
processing that this server might have done prior to losing its stable storage will not cause future difficulties.

8.6.1. Operation in RECOVER-WAIT State

The server MUST NOT be responsive in RECOVER-WAIT state.

8.6.2. Transition Out of RECOVER-WAIT State

Upon entry to RECOVER-WAIT state the server MUST start a timer whose expiration is set to a time equal to the time the server went down (if known) or the time the server started (if the down-time is unknown) plus the maximum-client-lead-time. When this timer expires, the server will transition into RECOVER-DONE state.

This is to allow any IP addresses that were allocated by this server prior to loss of its client binding information in stable storage to contact the other server or to time out.

If this is the first time this server has run failover -- as determined by the information received from the partner, not necessarily only as determined by this server’s stable storage (as that may have been lost), then the waiting time discussed above may be skipped, and the server may transition immediately to RECOVER-DONE state.

If the server has never before run failover, then there is no need to wait in this state -- but, again, to determine if this server has run failover it is vital that the information provided by the partner be utilized, since the stable storage of this server may have been lost.

If communications fails while a server is in RECOVER-WAIT state, it has no effect on the operation of this state. The server SHOULD continue to operate its timer, and the timer expires during the period where communications with the other server have failed, then the server SHOULD transition to RECOVER-DONE state. This is rare -- failover state transitions are not usually made while communications are interrupted, but in this case there is no reason to inhibit the timer.

8.7. RECOVER-DONE State

This state exists to allow an interlocked transition for one server from RECOVER state and another server from PARTNER-DOWN or COMMUNICATIONS-INTERRUPTED state into NORMAL state.
8.7.1. Operation in RECOVER-DONE State

A server in RECOVER-DONE state MUST respond only to DHCPREQUEST/RENEWAL and DHCPREQUEST/REBINDING DHCP messages.

8.7.2. Transition Out of RECOVER-DONE State

When a server in RECOVER-DONE state determines that its partner server has entered NORMAL or RECOVER-DONE state, then it will transition into NORMAL state.

If communications fails while in RECOVER-DONE state, a server will stay in RECOVER-DONE state.

8.8. NORMAL State

NORMAL state is the state used by a server when it is communicating with the other server, and any required resynchronization has been performed. While some bindings database synchronization is performed in NORMAL state, potential conflicts are resolved prior to entry into NORMAL state as is binding database data loss.

When entering NORMAL state, a server will send to the other server all currently unacknowledged binding updates as BNDUPD messages.

When the above process is complete, if the server entering NORMAL state is a secondary server, then it will request IP addresses for allocation using the POOLREQ message.

8.8.1. Operation in NORMAL State

When in NORMAL state a server will operate in the following manner:

Lease time calculations
As discussed in Section 7.4, the lease interval given to a DHCP client can never be more than the MCLT greater than the most recently received potential-expiration-time from the failover partner or the current time, whichever is later.

As long as a server adheres to this constraint, the specifics of the lease interval that it gives to a DHCP client or the value of the potential-expiration-time sent to its failover partner are implementation dependent.

Lazy update of partner server
After sending an REPLY that includes lease update to a client, the server servicing a DHCP client request attempts to update its partner with the new binding information. Server transmits both
desired valid lifetime and actual valid lifetime.

Reallocation of IP addresses between clients Whenever a client binding is released or expires, a BNDUPD message must be sent to the partner, setting the binding state to RELEASED or EXPIRED. However, until a BNDACK is received for this message, the IP address cannot be allocated to another client. It cannot be allocated to the same client again if a BNDUPD was sent, otherwise it can. See Section 7.6.

In normal state, each server receives binding updates from its partner server in BNDUPD messages. It records these in its client binding database in stable storage and then sends a corresponding BNDACK message to its partner server.

8.8.2. Transition Out of NORMAL State

If an external command is received by a server in NORMAL state informing it that its partner is down, then transition into PARTNER-DOWN state. Generally, this would be an unusual situation, where some external agency knew the partner server was down. Using the command in this case would be appropriate if the polling interval and timeout were long.

If a server in NORMAL state fails to receive acks to messages sent to its partner for an implementation dependent period of time, it MAY move into COMMUNICATIONS-INTERRUPTED state. This situation might occur if the partner server was capable of maintaining the TCP connection between the server and also capable of sending a CONTACT message every tSend seconds, but was (for some reason) incapable of processing BNDUPD messages.

If the communications is determined to not be "ok" (as defined in Section 7.5), then transition into COMMUNICATIONS-INTERRUPTED state.

If a server in NORMAL state receives any messages from its partner where the partner has changed state from that expected by the server in NORMAL state, then the server should transition into COMMUNICATIONS-INTERRUPTED state and take the appropriate state transition from there. For example, it would be expected for the partner to transition from POTENTIAL-CONFLICT into NORMAL state, but not for the partner to transition from NORMAL into POTENTIAL-CONFLICT state.

If a server in NORMAL state receives any messages from its partner where the PARTNER has changed into SHUTDOWN state, the server should transition into PARTNER-DOWN state.
8.9. COMMUNICATIONS-INTERRUPTED State

A server goes into COMMUNICATIONS-INTERRUPTED state whenever it is unable to communicate with its partner. Primary and secondary servers cycle automatically (without administrative intervention) between NORMAL and COMMUNICATIONS-INTERRUPTED state as the network connection between them fails and recovers, or as the partner server cycles between operational and non-operational. No duplicate IP address allocation can occur while the servers cycle between these states.

When a server enters COMMUNICATIONS-INTERRUPTED state, if it has been configured to support an automatic transition out of COMMUNICATIONS-INTERRUPTED state and into PARTNER-DOWN state (i.e., a "safe period" has been configured, see section 10), then a timer MUST be started for the length of the configured safe period.

A server transitioning into the COMMUNICATIONS-INTERRUPTED state from the NORMAL state SHOULD raise some alarm condition to alert administrative staff to a potential problem in the DHCP subsystem.

8.9.1. Operation in COMMUNICATIONS-INTERRUPTED State

In this state a server MUST respond to all DHCP client requests. When allocating new lease, each server allocates from its own pool, where the primary MUST allocate only FREE resources (addresses or prefixes), and the secondary MUST allocate only BACKUP resources (addresses or prefixes). When responding to RENEW messages, each server will allow continued renewal of a DHCP client’s current lease on an IP address or prefix irrespective of whether that lease was given out by the receiving server or not, although the renewal period MUST NOT exceed the maximum client lead time (MCLT) beyond the latest of: 1) the potential valid lifetime already acknowledged by the other server, or 2) the lease- expiration-time, or 3) the potential valid lifetime received from the partner server.

However, since the server cannot communicate with its partner in this state, the acknowledged potential valid lifetime will not be updated in any new bindings. This is likely to eventually cause the actual valid lifetimes to be the current time plus the MCLT (unless this is greater than the desired-client-lease- time).

The server should continue to try to establish a connection with its partner.
8.9.2. Transition Out of COMMUNICATIONS-INTERRUPTED State

If the safe period timer expires while a server is in the COMMUNICATIONS-INTERRUPTED state, it will transition immediately into PARTNER-DOWN state.

If an external command is received by a server in COMMUNICATIONS-INTERRUPTED state informing it that its partner is down, it will transition immediately into PARTNER-DOWN state.

If communications is restored with the other server, then the server in COMMUNICATIONS-INTERRUPTED state will transition into another state based on the state of the partner:

- NORMAL or COMMUNICATIONS-INTERRUPTED: Transition into the NORMAL state.
- RECOVER: Stay in COMMUNICATIONS-INTERRUPTED state.
- RECOVER-DONE: Transition into NORMAL state.
- PARTNER-DOWN, POTENTIAL-CONFLICT, CONFLICT-DONE, or RESOLUTION-INTERRUPTED: Transition into POTENTIAL-CONFLICT state.
- SHUTDOWN: Transition into PARTNER-DOWN state.

The following figure illustrates the transition from NORMAL to COMMUNICATIONS-INTERRUPTED state and then back to NORMAL state again.
Figure 3: Transition from NORMAL to COMMUNICATIONS-INTERRUPTED and back (example with 2 addresses allocated to secondary)

8.10. POTENTIAL-CONFLICT State

This state indicates that the two servers are attempting to reintegrate with each other, but at least one of them was running in a state that did not guarantee automatic reintegration would be possible. In POTENTIAL-CONFLICT state the servers may determine that
the same resource has been offered and accepted by two different clients.

It is a goal of this protocol to minimize the possibility that POTENTIAL-CONFLICT state is ever entered.

When a primary server enters POTENTIAL-CONFLICT state it should request that the secondary send it all updates of which it is currently unaware by sending an UPDREQ message to the secondary server.

A secondary server entering POTENTIAL-CONFLICT state will wait for the primary to send it an UPDREQ message.

8.10.1. Operation in POTENTIAL-CONFLICT State

Any server in POTENTIAL-CONFLICT state MUST NOT process any incoming DHCP requests.

8.10.2. Transition Out of POTENTIAL-CONFLICT State

If communications fails with the partner while in POTENTIAL-CONFLICT state, then the server will transition to RESOLUTION-INTERRUPTED state.

Whenever either server receives an UPDDONE message from its partner while in POTENTIAL-CONFLICT state, it MUST transition to a new state. The primary MUST transition to CONFLICT-DONE state, and the secondary MUST transition to NORMAL state. This will cause the primary server to leave POTENTIAL-CONFLICT state prior to the secondary, since the primary sends an UPDREQ message and receives an UPDDONE before the secondary sends an UPDREQ message and receives its UPDDONE message.

When a secondary server receives an indication that the primary server has made a transition from POTENTIAL-CONFLICT to CONFLICT-DONE state, it SHOULD send an UPDREQ message to the primary server.
### 8.11. RESOLUTION-INTERRUPTED State

This state indicates that the two servers were attempting to reintegrate with each other in POTENTIAL-CONFLICT state, but communications failed prior to completion of re-integration.

If the servers remained in POTENTIAL-CONFLICT while communications was interrupted, neither server would be responsive to DHCP client requests, and if one server had crashed, then there might be no server able to process DHCP requests.
When a server enters RESOLUTION-INTERRUPTED state it SHOULD raise an alarm condition to alert administrative staff of a problem in the DHCP subsystem.

8.11.1. Operation in RESOLUTION-INTERRUPTED State

In this state a server MUST respond to all DHCP client requests. When allocating new resources (addresses or prefixes), each server SHOULD allocate from its own pool (if that can be determined), where the primary SHOULD allocate only FREE resources, and the secondary SHOULD allocate only BACKUP resources. When responding to renewal requests, each server will allow continued renewal of a DHCP client’s current lease irrespective of whether that lease was given out by the receiving server or not, although the renewal period MUST not exceed the maximum client lead time (MCLT) beyond the latest of: 1) the potential valid lifetime already acknowledged by the other server or 2) the lease-expiration-time or 3) potential valid lifetime received from the partner server.

However, since the server cannot communicate with its partner in this state, the acknowledged potential valid lifetime will not be updated in any new bindings.

8.11.2. Transition Out of RESOLUTION-INTERRUPTED State

If an external command is received by a server in RESOLUTION-INTERRUPTED state informing it that its partner is down, it will transition immediately into PARTNER-DOWN state.

If communications is restored with the other server, then the server in RESOLUTION-INTERRUPTED state will transition into POTENTIAL-CONFLICT state.

8.12. CONFLICT-DONE State

This state indicates that during the process where the two servers are attempting to re-integrate with each other, the primary server has received all of the updates from the secondary server. It make a transition into CONFLICT-DONE state in order that it may be totally responsive to the client load, as opposed to NORMAL state where it would be in a "balanced" responsive state, running the load balancing algorithm.

TODO: We do not support load balancing, so CONFLICT-DONE is actually equal to NORMAL. Need to remove CONFLICT-DONE and replace all its references to NORMAL.
8.12.1. Operation in CONFLICT-DONE State

A primary server in CONFLICT-DONE state is fully responsive to all DHCP clients (similar to the situation in COMMUNICATIONS-INTERRUPTED state).

If communications fails, remain in CONFLICT-DONE state. If communications becomes OK, remain in CONFLICT-DONE state until the conditions for transition out become satisfied.

8.12.2. Transition Out of CONFLICT-DONE State

If communications fails with the partner while in CONFLICT-DONE state, then the server will remain in CONFLICT-DONE state.

When a primary server determines that the secondary server has made a transition into NORMAL state, the primary server will also transition into NORMAL state.

8.13. PAUSED State

TODO: Remove PAUSED state completely

This state exists to allow one server to inform another that it will be out of service for what is predicted to be a relatively short time, and to allow the other server to transition to COMMUNICATIONS-INTERRUPTED state immediately and to begin servicing all DHCP clients with no interruption in service to new DHCP clients.

A server which is aware that it is shutting down temporarily SHOULD send a STATE message with the server-state option containing PAUSED state and close the TCP connection.

While a server may or may not transition internally into PAUSED state, the 'previous' state determined when it is restarted MUST be the state the server was in prior to receiving the command to shut-down and restart and which precedes its entry into the PAUSED state. See Section 8.3.2 concerning the use of the previous state upon server restart.

When entering PAUSED state, the server MUST store the previous state in stable storage, and use that state as the previous state when it is restarted.

8.13.1. Operation in PAUSED State

Server MUST NOT perform any operation while in PAUSED state.
8.13.2. Transition Out of PAUSED State

A server makes a transition out of PAUSED state by being restarted. At that time, the previous state MUST be the state the server was in prior to entering the PAUSED state.

8.14. SHUTDOWN State

This state exists to allow one server to inform another that it will be out of service for what is predicted to be a relatively long time, and to allow the other server to transition immediately to PARTNER-DOWN state, and take over completely for the server going down.

When entering SHUTDOWN state, the server MUST record the previous state in stable storage for use when the server is restarted. It also MUST record the current time as the last time operational.

A server which is aware that it is shutting down SHOULD send a STATE message with the server-state field containing SHUTDOWN.

8.14.1. Operation in SHUTDOWN State

A server in SHUTDOWN state MUST NOT respond to any DHCP client input.

If a server receives any message indicating that the partner has moved to PARTNER-DOWN state while it is in SHUTDOWN state then it MUST record RECOVER state as the previous state to be used when it is restarted.

A server SHOULD wait for a few seconds after informing the partner of entry into SHUTDOWN state (if communications are okay) to determine if the partner entered PARTNER-DOWN state.

8.14.2. Transition Out of SHUTDOWN State

A server makes a transition out of SHUTDOWN state by being restarted.

9. Proposed extensions

The following section discusses possible extensions to the proposed failover mechanism. Listed extensions must be sufficiently simple to not further complicate failover protocol. Any proposals that are considered complex will be defined as stand-alone extensions in separate documents.
9.1. Active-active mode

A very simple way to achieve active-active mode is to remove the restriction that secondary server MUST NOT respond to SOLICIT and REQUEST messages. Instead it could respond, but MUST have lower preference than primary server. Clients discovering available servers will receive ADVERTISE messages from both servers, but are expected to select the primary server as it has higher preference value configured. The following REQUEST message will be directed to primary server.

Discussion: Do DHCPv6 clients actually do this? DHCPv4 clients were rumored to wait for a "while" to accept the best offer, but to a first approximation, they all take the first offer they receive that is even acceptable.

The benefit of this approach, compared to the "basic" active--passive solution is that there is no delay between primary failure and the moment when secondary starts serving requests.

Discussion: The possibility of setting both servers preference to an equal value could theoretically work as a crude attempt to provide load balancing. It wouldn’t do much good on its own, as one (faster) server could be chosen more frequently (assuming that with equal preference sets clients will pick first responding server, which is not mandated by DHCPv6). We could design a simple mechanism of dynamically updating preference depending on usage of available resources. This concept hasn’t been investigated in detail yet.

10. Dynamic DNS Considerations

TODO: Describe DNS Updates challenges in failover environment. It is nicely described in Section 5.12 of [dhcv4-failover].

11. Reservations and failover

TODO: Describe how lease reservation works with failover. See Section 5.13 in [dhcv4-failover].

12. Protocol entities

Discussion: It is unclear if following sections belong to design or protocol draft. It is currently kept here as a scratchbook with list of things that will have to be defined eventually. Whether or not it will stay in this document or will be moved to the protocol spec
12.1. Failover Protocol

This section enumerates list of options that will be defined in failover protocol specification. Rough description of purpose and content for each option is specified. Exact on wire format will be defined in protocol specification.

1. OPTION_FO_TIMESTAMP - convey information about timestamp. It is used by time skew measurement algorithm (see Section 7.1).

12.2. Protocol constants

This section enumerates various constants that have to be defined in actual protocol specification.

1. TIME_SKEW_PKTS_AVG - number of packets that are used to calculate average time skew between partners. See (see Section 7.1).

13. Open questions

This is scratchbook. This section will be removed once questions are answered.

Q: Do we want to support temporary addresses? I think not. They are short-lived by definition, so clients should not mind getting new temporary addresses.

Q: Do we want to support CGA-registered addresses? There is currently work in DHC WG about this, but I haven’t looked at it yet. If that is complicated, we may not define it here, but rather as an extension. [If it moves forward, we need to support it.]

14. Security Considerations

TODO: Security considerations section will contain loose notes and will be transformed into consistent text once the core design solidifies.

15. IANA Considerations

IANA is not requested to perform any actions at this time.
16. Acknowledgements

This document extensively uses concepts, definitions and other parts of [dhcpv4-failover] document. Authors would like to thank Shawn Routher, Greg Rabil, and Bernie Volz for their significant involvement and contributions.

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).

17. References

17.1. Normative References


17.2. Informative References


Authors’ Addresses

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

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

Kim Kinnear
Cisco Systems, Inc.
1414 Massachusetts Ave.
Boxborough, Massachusetts 01719
USA

Phone: +1 (978) 936-0000
Email: kkinnear@cisco.com
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.
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 an 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 IA_PREFIX (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


10.2. Informative References

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
Abstract

The DHCPv6 RADIUS option provides a communication mechanism between relay agent and the server. This mechanism can help the centralized DHCPv6 server to select the right configuration for the client based on the authorization information received from a separate RADIUS server which is not located at the same place of DHCPv6 server in the cases where the NAS acts as DHCPv6 relay agent and RADIUS client simultaneously.

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 October 27, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents.
carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction .................................................. 3
2. Terminology and Language ..................................... 3
3. Network Scenarios .............................................. 4
4. OPTION_RADIUS .................................................. 6
5. Relay Agent Behavior .......................................... 6
6. Server Behavior ................................................ 7
7. Client Behavior ................................................ 7
8. Security Considerations ....................................... 7
9. IANA Considerations .......................................... 7
10. Acknowledgements ............................................ 7
11. References .................................................... 7
    11.1. Normative References .................................... 7
    11.2. Informative References .................................. 8
Authors' Addresses .............................................. 8
1. Introduction

DHCPv6 provides a mechanism that allows the server to assign or delegate both stateful and stateless configuration parameters to the clients. The stateful configuration parameters include IPv6 address [RFC3315], IPv6 prefix [RFC3633], and etc. The stateless configuration parameters [RFC3736] include, for example, DNS [RFC3646]. The DHCPv6 server is typically deployed in the central part of an ISP network.

RADIUS [RFC2865], an essentially stateless protocol, is used widely as the centralized authentication, authorization and user management mechanism for the service provision in Broadband access network. [RFC3162], [RFC4818] and [ietf-radext-ipv6-access-06] specify attributes that supports the provision of service for IPv6 access. RADIUS authorizes that the NAS assigns an IPv6 address or prefix from the indicated pool, or assigns an IPv6 address or prefix with an explicitly indicated value in the attributes for the subscribers.

These mechanisms work well in the deployment scenario where the NAS acts as the distributed DHCPv6 server. In this case the NAS responds as the indication conveyed by the attributes in the Access-Accept message from the RADIUS server. These mechanisms also work in the scenario where the centralized DHCPv6 server is co-located with the RADIUS server, where they can share the same database of the users. But when the NAS acts as the relay agent and RADIUS client simultaneously, and the centralized DHCPv6 server is not located in the same place as the RADIUS server, a new communication mechanism is needed for the relay agent to transfer the authorization information indicated by the RADIUS attributes to the DHCPv6 server.

2. Terminology and Language

This document specifies a DHCPv6 option for the distributed Relay Agent to transfer the authorization information of RADIUS attributes received in the Access-Accept message to the centralized DHCPv6 server. This document should be read in conjunction with the following specifications: [RFC2865], [RFC2869], [RFC3315] and [RFC4818]. These specifications will help the reader to understand how DHCPv6 and RADIUS work together to provide IPv6 service. Definitions for terms and acronyms not specified in this document are defined in [RFC2865], [RFC2869], [RFC3315] and [RFC4818].

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in BCP 14, [RFC2119].
3. Network Scenarios

Figure 1 and Figure 2 shows the typical network scenarios where the communication mechanism introduced in this document is necessary. In these scenarios, the centralized DHCPv6 server is not co-located with the RADIUS server, but both of them are in the same administrative domain. The NAS acts as the relay agent and the RADIUS client simultaneously. Figure 1 shows the sequence of DHCPv6 and RADIUS messages for IPoE access mode. Figure 2 shows the sequence of DHCPv6 and RADIUS messages for PPPoE access mode.

Figure 1: Network scenario and message sequence when employing DHCPv6
If the authorization through RADIUS fails, the associated message sequences will stop. The DHCPv6 relay will not forward the message.
from the client to the server.

4. OPTION_RADIUS

The OPTION_RADIUS is a stateless DHCPv6 option, and is used by the relay agent to carry the authorization information of RADIUS attributes received in the Access-Accept message.

The format of the OPTION_RADIUS option is defined as follows:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         OPTION_RADIUS         |         option-length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       RADIUS Attributes...                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- **option-code**: TBD
- **option-length**: Length of the option-data in Octets
- **option-data**: One or a list of RADIUS Attributes

The option-data of OPTION_RADIUS is one or a list of RADIUS attributes received in the Access-Accept message from the RADIUS server. As the same method in [RFC4014], only the attributes listed in the table below may be included in the OPTION_RADIUS.

<table>
<thead>
<tr>
<th>Type Code</th>
<th>Attribute</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>Vendor-Specific</td>
<td>[RFC2865]</td>
</tr>
<tr>
<td>88</td>
<td>Framed-Pool</td>
<td>[RFC2869]</td>
</tr>
<tr>
<td>123</td>
<td>Delegated-IPv6-Prefix</td>
<td>[RFC4818]</td>
</tr>
<tr>
<td>[TBD]</td>
<td>Framed-IPv6-Address</td>
<td>[ietf-radext-ipv6-access-06]</td>
</tr>
</tbody>
</table>

Note: The above table might have more attributes in the future.

5. Relay Agent Behavior

The DHCPv6 relay agent may include OPTION_RADIUS in the RELAY-FORW message. When the value in the attributes of Framed-Pool (88), (or Stateful-IPv6-Address-Pool, Delegated-IPv6-Prefix-Pool,) Delegated-IPv6-Prefix (123) and Framed-IPv6-Address in the Access-Accept message replied from RADIUS server are valid, the relay agent that supports OPTION_RADIUS SHOULD include these RADIUS attributes in the container option, OPTION_RADIUS. The relay agent MUST ignore OPTION_RADIUS if received.
6. Server Behavior

Upon receipt of the RELAY-FORW message with OPTION_RADIUS from a relay agent, the DHCPv6 server SHOULD extract and interpret the RADIUS attributes in the OPTION_RADIUS, and use that information in selecting configuration parameters for the requesting client. If the DHCPv6 server does not support OPTION_RADIUS, the DHCPv6 server SHOULD ignore this option. The DHCPv6 server MUST NOT include OPTION_RADIUS in RELAY-REPL messages.

7. Client Behavior

OPTION_RADIUS option is only exchanged between the relay agents and the servers. DHCPv6 clients are not aware of the usage of OPTION_RADIUS. DHCPv6 Client MUST NOT send OPTION_RADIUS, and MUST ignore OPTION_RADIUS if received.

8. Security Considerations

Known security vulnerabilities of the DHCPv6 and RADIUS protocol may apply to its options. Security issues related with DHCPv6 are described in section 23 of [RFC3315]. Security issues related with RADIUS are described in section 8 of [RFC2865], section 5 of [RFC3162].

9. IANA Considerations

The authors of this document request to assign a new DHCPv6 option code for OPTION_RADIUS.

10. Acknowledgements

Expert comments from Bernie Volz and Tomek Mrugalski for the discussion on the technology selection in the mailing list are appreciated.

11. References

11.1. Normative References


11.2. Informative References


Authors’ Addresses

Leaf Y. Yeh (editor)
Huawei Technologies
P. R. China

Email: leaf.y.yeh@huawei.com

Mohamed Boucadair
France Telecom
France

Email: mohamed.boucadair@orange.com

Ted Lemon
Nominum, Inc
USA

Email: Ted.Lemon@nominum.com